RÉPUBLIQUE FRANÇAISE
Liberté Égalité Fraternité

Université de Toulouse

Doctorat de l'Université de Toulouse

préparé à Toulouse INP

Analyse différentielle de données Hi-C : évaluation d'outils et développement d'une méthode statistique reposant sur l'inférence post hoc

---

Thèse présentée et soutenue, le 21 octobre 2026 par

Elise JORGE

École doctorale
SEVAB - Sciences Ecologiques, Vétérinaires, Agronomiques et Bioingenieries

Spécialité
Infectiologie, Physiopathologie, Toxicologie, Génétique et Nutrition

Unité de recherche
GenPhySE- Unité Génétique, Physiologie et Systèmes d'Elevage

Thèse dirigée par
Sylvain FOISSAC et Nathalie VIALANEIX

Composition du jury
Mme Cathy MAUGIS-RABUSSEAU, Présidente, INSA Toulouse
Mme Patricia THÉBAULT, Rapporteure, Université de Bordeaux
M. Vincent BRAULT, Rapporteur, Université Grenoble Alpes
M. Julien CHIQUET, Examinateur, INRAE Île-de-France - Versailles-Saclay
M. Thomas DERRIEN, Examinateur, CNRS Bretagne et Pays de la Loire
M. Sylvain FOISSAC, Directeur de thèse, INRAE Occitanie-Toulouse
Mme Nathalie VIALANEIX, Co-directrice de thèse, INRAE Occitanie-Toulouse

Membres invités
M. Pierre Neuvial, Co-encadrant, CNRS Occitanie Ouest

Remerciements

En premier lieu, je tiens à adresser mes remerciements à mes deux rapporteurices, Vincent Brault et Patricia Thébault. Merci d'avoir accepté de consacrer du temps à l'évaluation de mon travail. De même, je tiens à remercier Julien Chiquet, Thomas Derrien et Cathy Maugis-Rabusseau d'avoir accepté d'être membres de mon jury. En particulier, merci Cathy de m'avoir donné l'opportunité de découvrir la recherche durant mon stage de M2 et de m'avoir motivée à faire cette thèse.

Je tiens également à remercier les trois membres de mon CSI, Marie Chavent, Anne-Laure Valton et Nelle Varoquaux. Vos retours objectifs et votre bienveillance m'ont permis de prendre confiance au fur et à mesure de ma thèse.

J'adresse ma sincère gratitude à mes trois encadrantes, Nathalie, Pierre et Sylvain. Nathalie, merci pour ta pédagogie, ta rigueur, ta confiance et ton attention à mon égard. Pierre, tout d'abord merci d'avoir été le premier à venir me parler de cette thèse ! Merci également pour ta disponibilité et ton implication dans l'encadrement de ma thèse malgré la visio. Sylvain, merci pour toutes les explications de concepts bio(logiques/informatiques), merci surtout pour tes encouragements et ta bonne humeur ! J'ai la sensation d'avoir été entourée de trois personnes très compétentes et d'avoir beaucoup progressé grâce à vous.

Merci au laboratoire GenPhySE de m'avoir accueillie pendant ces trois années. De l'équipe DynaGen à REGLISS, je n'ai compris qu'une fraction des échanges scientifiques auxquels j'ai assisté mais j'ai toujours eu l'impression d'être entourée de personnes compétentes et passionnées par leur métier. Je remercie également le laboratoire MIAT – de l'autre côté du couloir – de m'avoir permis de garder un lien avec la statistique.

À toutes les organisatrices du programme Agreenium EIR-A, merci pour votre dévouement. Ces deux séminaires, qui avaient des airs de colonies de vacances studieuses, font partie des meilleurs moments de ma thèse.

Je tiens à remercier toutes les acteurices de mon séjour au Canada. Merci Catherine de t'être occupée du casse-tête administratif de la mission. Merci aux départements GA et MathNum, à l'INP Toulouse, au métaprogramme DIGIT-BIO et au laboratoire GenPhySE d'avoir financé ce séjour. Toby, merci de m'avoir chaleureusement accueillie à Sherbrooke et de m'avoir aidée à aborder des perspectives de travail stimulantes. À mes collègues sherbrookoises, merci pour votre accueil et nos pauses-déjeuner aux airs de podcasts !

Dominique Pantalacci, je vous remercie pour votre disponibilité et votre réactivité, de l'inscription en thèse à la soutenance.

Aux deux stagiaires que j'ai eu la chance d'encadrer, Roman et Bastian, merci, c'étaient deux super expériences de travailler avec vous. À toutes les autres stagiaires dont j'ai croisé la route, Océane, Axelle, Emma, Margaux, Justine, etc., merci d'avoir apporté de la joie au labo durant vos quelques mois ici !

À mes collègues doctorantes, Suzanne, Han, Noémien, Hélène, Julie, Stacy, etc., ça a été un grand plaisir de partager ces années de thèse plus ou moins en simultané avec vous. Bon courage à celles qui resteront ! Han, je reviendrai te faire coucou régulièrement :) Artchure, fidèle membre de la gang, on n'a peut-être pas écrit les meilleures thèses, mais en termes de conneries, on est plutôt pas mal. Alice, je me sens très chanceuse d'avoir pu partager mon bureau avec toi ! J'aurais pu être impressionnée par tes connaissances qui me semblent infinies sur n'importe quel sujet mais ça ne m'a pas empêchée de t'abreuver de ma logorrhée motivée par à peu près toutes mes contrariétés.

Élodie, merci de m'avoir aidée à respirer durant la première moitié de ma thèse ! Merci aussi de t'être impliquée dans le soutien aux doctorantes.

Clarisse, merci pour votre bienveillance, votre honnêteté et vos conseils avisés.

À mes amis, merci ! Sarah, de la L2 à maintenant, merci pour nos rires et nos râleries.

À ma famille, Papa, Maman, Joanito, Titi, merci de m'avoir soutenue dans la poursuite de mes études. Je dois vous confirmer, comme aime le dire Papa, que j'ai effectivement poursuivi mes études sans jamais les rattraper... [1]

Mon Jojo, merci pour tout, merci d'être unique. [2]

À toutes les personnes que j'ai oubliées – sciemment ou non – et comme je ne suis pas avare, je vous laisse ce "merci" que vous pourrez vous attribuer à votre guise.

Enfin, avec toute la modestie qui me constitue, je me remercie d'avoir été jusqu'au bout de cette thèse et de sa rédaction.

1. En toute rigueur bibliographique, je me dois de préciser que cette citation est attribuée à Alphonse Allais.
2. (Six Seven)

«Te iterum vidi
In loco unde tota Massilia conspicitur
Oculis tuis tam nigris quam mea Twingo
Risuque tuo petulanti.»

Iulianus Mari

Résumé

De par la contrainte spatiale imposée par la taille du noyau dans lequel il est contenu, le génome se retrouve replié en trois dimensions (3D). Ce repliement induit la formation de structures organisées de manière hiérarchique. De ce fait, il est nécessaire d'aller au-delà d'une vision linéaire du génome pour comprendre le lien entre structures tridimensionnelles et fonctions associées. La technique Hi-C permet de cartographier l'organisation 3D du génome. L'analyse différentielle des données Hi-C, c'est-à-dire la comparaison des données issues de cette expérience entre plusieurs conditions biologiques, peut mettre en évidence des réarrangements dans la conformation spatiale du génome. Les résultats de ce type d'analyse contribuent à comprendre l'impact des réorganisations spatiales sur des processus tels que la régulation de l'expression des gènes ou la différenciation cellulaire.

Dans une première partie de la thèse, nous présentons un travail de comparaison de méthodes d'analyse différentielle de données Hi-C basées sur les pixels. Ce type de méthode effectue, pour chaque paire de positions génomiques, un test statistique mesurant la différence d'interaction entre les deux conditions biologiques étudiées. Nous avons réalisé une typologie des outils en fonction d'aspects techniques (implémentation, format de données en entrée, etc.) ainsi que méthodologiques (filtrages, normalisations, type de modèle statistique utilisé, type de prise en compte de la multiplicité des tests, etc.). Ensuite, nous nous sommes intéressés à l'évaluation des différents outils sur un jeu de données ne contenant pas de pixels différentiels, permettant de vérifier le contrôle correct de l'erreur de type I. Puis, nous avons évalué ces outils sur des données contenant des pixels différentiels et dont la position est connue, afin d'évaluer l'erreur de type II et donc la puissance de détection des outils. Cette comparaison a permis de mettre en avant les performances supérieures de la méthode **diffHiC** sur le contrôle de l'erreur de type I et la puissance de détection, par rapport aux autres outils.

Dans une seconde partie, nous présentons une nouvelle approche d'analyse différentielle – **hicream** – visant à identifier des régions différentielles de forme arbitraire. Cette approche a été motivée par les limitations des méthodes existantes d'analyse différentielle de données Hi-C. D'une part, les approches d'analyse différentielle basées sur les structures ne permettent pas de découvrir de nouvelles régions différentielles et ne fournissent pas toujours de garanties statistiques. D'autre part, les méthodes d'analyse basées sur les pixels produisent des résultats souvent peu interprétables avec des détections différentielles ponctuelles qui ne peuvent être généralisées à l'identification de régions différen-

tielles. La méthode **hicream** est entièrement guidée par les données car elle définit les régions candidates au moyen d'un clustering. Ensuite, grâce aux résultats des tests fournis par **diffHiC**, elle s'appuie sur l'inférence *post hoc* pour évaluer chaque région candidate et fournir une proportion minimale de pixels différentiels dans chacune d'entre elles, permettant ainsi d'identifier les plus différentielles. La pertinence de la méthode a été démontrée à la fois sur des données semi-simulées et sur des données réelles en lien avec des signaux biologiques extérieurs. D'autre part, en comparant nos résultats à des outils d'identification de structures différentielles, nous avons pu montrer que nous pouvions trouver des clusters différentiels qui correspondent à des structures connues dans les données Hi-C telles que des TADs (*Topologically Associating Domains*) ou des compartiments. Enfin, l'interprétation des clusters différentiels détectés, de formes arbitraires, est facilitée par le développement d'une visualisation interactive des résultats.

Abstract

Due to spatial constraints imposed by the size of the nucleus in which it is contained, the genome is folded in three dimensions (3D). This folding leads to the formation of hierarchically organized structures. As a result, it is necessary to move beyond a linear view of the genome to understand the relationship between three-dimensional structures and their associated functions. The Hi-C technique allows for the mapping of the genome's 3D organization. Differential analysis of Hi-C data – that is, comparing data from the experiment across multiple biological conditions – can reveal rearrangements in the genome's spatial conformation. The results of this analysis contribute to understanding the impact of spatial reorganizations on biological processes such as gene expression regulation or cell differentiation.

In the first part of my thesis, we present a benchmark study of pixel-based differential analysis methods for Hi-C data. This type of method performs, for each pair of genomic positions, a statistical test measuring the difference in interaction between the two biological conditions under study. We provided a typology of the tools in terms of their technical aspects (implementation, input data format, etc.) as well as their methodological aspects (filtering, normalization, type of statistical model used, type of multiple testing correction, etc.). Next, we evaluated the various tools on a dataset with no differential pixels to assess type I error control. Then, we evaluated the tools on a dataset containing differential pixels with known positions in order to assess type II error and thus the tools' detection power. This benchmark highlighted the superior performance of the **diffHiC** method in type I error control and in detection power, compared with the other tools.

In the second part, we present a new differential analysis approach – **hicream** – aimed at identifying differential regions of arbitrary shapes. This approach was motivated by the limitations of existing Hi-C data differential analysis methods. On the one hand, structure-based differential analysis approaches do not allow for the discovery of new differential regions and do not always provide statistical guarantees. On the other hand, pixel-based analysis methods often produce results that are difficult to interpret, with point-wise differential detections that cannot be generalized to the identification of differential regions. The **hicream** method is entirely data-driven, as it defines candidate regions using clustering. Then, using the tests results provided by **diffHiC**, it relies on *post hoc* inference to evaluate each candidate region and provide a minimum proportion of differential pixels within each, thereby identifying the most differential regions. The effectiveness of the me-

thod was demonstrated on both semi-simulated data and real data, in relation to external biological signals. Furthermore, by comparing our results to structure-based differential analysis results, we were able to demonstrate that we could identify differential clusters that correspond to differential structures in Hi-C data, such as TADs (*Topologically Associating Domains*) or compartments. Finally, the interpretation of the detected differential clusters was facilitated by the development of an interactive visualization tool, which also highlighted the shape diversity.

Table des matières

1 Introduction **1**
1.1 Contexte biologique . . . 1
1.2 Caractérisation de la structure 3D du génome et données Hi-C . . . 4
1.2.1 Expérience Hi-C . . . 5
1.2.2 Pré-traitements et normalisations des données Hi-C . . . 6
1.2.3 Structures tridimensionnelles hiérarchiques . . . 8
1.2.4 Perturbations de structures et rôle fonctionnel . . . 12
1.3 Analyse différentielle de données Hi-C . . . 13
1.3.1 Définition mathématique . . . 15
1.3.2 Analyse différentielle basée sur les pixels . . . 15
1.3.3 Analyse différentielle basée sur les structures . . . 16
1.4 Tests statistiques . . . 18
1.4.1 Définition et construction d'un test statistique . . . 18
1.4.2 Validité d'un test et contrôle de l'erreur . . . 18
1.4.3 Tests multiples . . . 19
1.4.4 Application à l'analyse différentielle de données Hi-C . . . 20
1.5 Inférence *post hoc* . . . 21
1.6 Contributions de la thèse . . . 23

2 Analyse et évaluation des outils d'analyse différentielle de données Hi-C basée sur les pixels **25**

3 hicream : méthode d'identification de régions différentielles dans les données Hi-C **43**

4 Conclusions et perspectives **55**

Annexes **63**

Bibliographie **101**

Chapitre 1

Introduction

1.1 Contexte biologique

Le génome

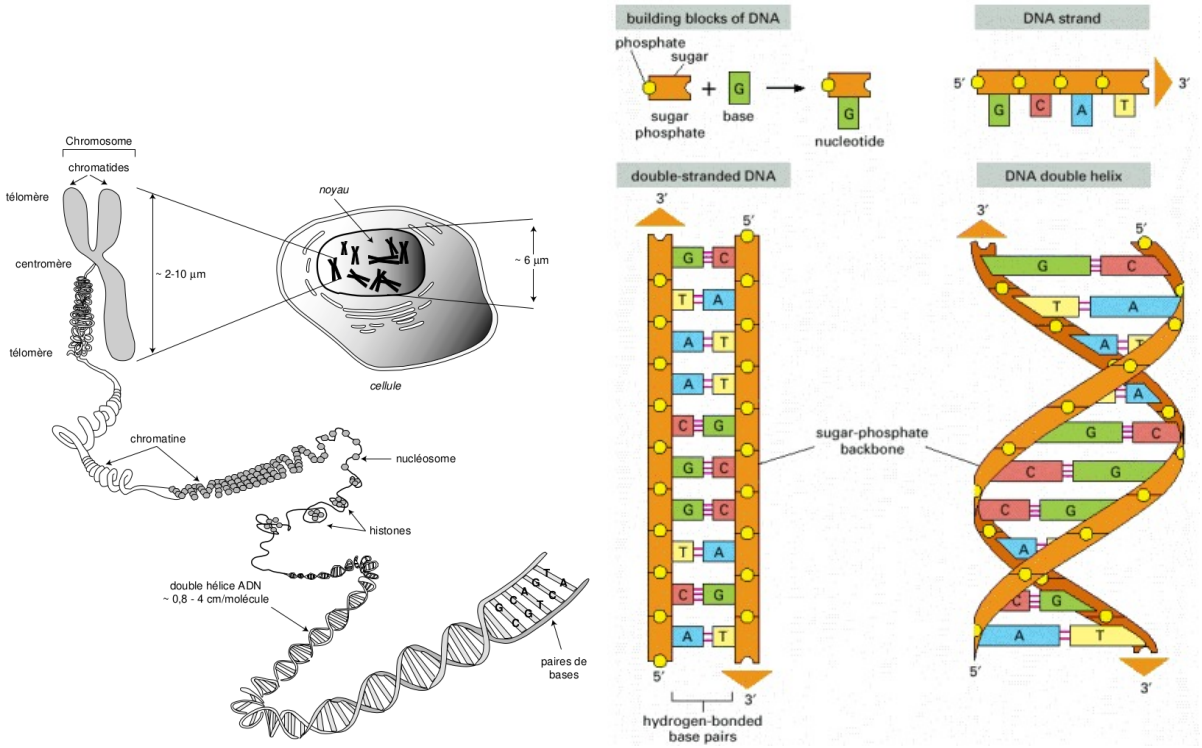


FIGURE 1.1 – Gauche : Schéma de la compaction de l'ADN en chromosome au sein du noyau de la cellule. Figure issue de Hayes [2000]. Droite : Schéma de la composition de l'ADN double brin. Figure issue de Alberts et al. [2002].

Le génome est l'ensemble de l'information génétique d'un organisme. Chez les eucaryotes, cette information est contenue dans le noyau de chaque cellule sous la forme de

chromosomes. Comme cela peut être observé dans la partie gauche de la figure 1.1, les chromosomes – dont la taille et le nombre standard varient selon l'espèce – contiennent l'information génétique d'un organisme à travers l'ADN. Plus précisément, au sein du chromosome, la fibre d'ADN est empaquetée autour de protéines – les histones – en une structure appelée chromatine.

Comme illustré dans la partie droite de la figure 1.1, l'ADN (acide désoxyribonucléique) est organisé en deux brins complémentaires antiparallèles formant une double hélice. Chaque brin d'ADN est un enchaînement de nucléotides qui sont constitués d'un phosphate, d'un sucre et d'une base. On retrouve quatre bases différentes : l'adénine (A), la guanine (G), la thymine (T) et la cytosine (C). Les deux brins d'ADN sont parallèles, orientés en sens contraires et leur complémentarité implique qu'une base (A) sur un brin est complétée par une base (T) sur l'autre brin tandis qu'une base (C) doit être complétée par une base (G). Les deux brins sont joints par des liaisons hydrogène entre chaque paire de bases. Des contraintes physico-chimiques mènent à la configuration la plus stable de l'ADN, en une double hélice. Afin de mesurer la longueur d'une portion d'ADN ou de chromatine, on utilise l'unité de la paire de bases (bp).

L'organisation 3D du génome

Les questions et problématiques abordées dans cette thèse découlent de l'observation suivante : une fois déplié, l'ADN contenu dans une cellule humaine mesure près de deux mètres tandis que le noyau de la cellule ne mesure lui que quelques micromètres [Hayes, 2000]. Ainsi, cette contrainte spatiale provoque un repliement des chromosomes sur eux-mêmes en trois dimensions, induisant des proximités spatiales entre des positions génomiques linéairement éloignées sur le chromosome.

La compaction de l'ADN joue un rôle important dans de nombreux processus biologiques tels que la différenciation cellulaire. On évoque, dans la suite, le rôle de cette organisation tridimensionnelle dans un processus sous-jacent essentiel à la réalisation des fonctions des cellules : l'expression des gènes.

L'expression des gènes

Le génome d'un individu est unique et, ainsi, l'information génétique portée par ses chromosomes au sein de l'ADN est la même dans toutes ses cellules. Cependant, toutes les cellules n'ont pas les mêmes fonctions. En effet, une cellule du pancréas peut avoir comme fonction de sécréter de l'insuline tandis qu'une cellule musculaire permet la contraction du muscle. On dit alors que ces cellules appartiennent à des types cellulaires différents. Le processus qui permet, sous certaines conditions, à une cellule souche (non-spécialisée) de se spécialiser en un type particulier est appelé différentiation cellulaire.

C'est le mécanisme de la régulation de l'expression des gènes qui permet alors de convertir la même information génétique en des fonctions cellulaires différentes. Le concept de gène a changé continuellement au cours du temps et il apparaît délicat d'en donner une définition scientifique consensuelle, simple et adaptée à tous les contextes dans lesquels cette notion est utilisée [Portin and Wilkins, 2017]. Dans le dictionnaire Larousse, un gène

est défini comme étant un : « *Segment d'A.D.N. conditionnant la synthèse d'une ou de plusieurs protéines et, donc, la manifestation et la transmission d'un caractère héréditaire déterminé.* ».

L'expression génique est le processus qui, à partir de la lecture d'un gène (un enchaînement spécifique de bases le long de l'ADN), permet la production de molécules remplissant des fonctions particulières au sein de la cellule. La première étape de l'expression d'un gène, appelée *transcription*, consiste en sa lecture par une ARN polymérase (une enzyme) puis sa copie complémentaire en ARN messager. Un promoteur est une séquence d'ADN située en amont d'un gène et qui permet à l'ARN polymérase de s'accrocher à la séquence. La lecture d'un gène est donc initiée dans sa région promotrice par l'ARN polymérase qui, au cours de l'étape d'élongation, parcourt le gène en « séparant » les deux brins. Chaque base du brin lu est copiée par son complémentaire – (C) en (G) et réciproquement, (T) en (A) – à ceci près qu'une base d'adénine (A) est copiée par une base d'uracile (U). Lorsque la copie du gène est terminée, on obtient un ARN messager dit transcrit de gène. L'étape suivante, qui permet d'obtenir une protéine à partir d'un ARN messager, est appelée *traduction*. La lecture de l'ARN messager par un ribosome s'effectue en dehors du noyau de la cellule au cours d'une étape d'élongation consistant à associer à des codons (groupes de 3 nucléotides successifs) l'acide aminé correspondant déterminé par le code génétique. La lecture de certains codons entraîne l'arrêt de la traduction, finalisant ainsi la production d'une chaîne d'acides aminés. Cette chaîne d'acides aminés, en fonction de son repliement tridimensionnel, devient alors une protéine qui possède des fonctions spécifiques au sein de la cellule. L'étape de traduction ne concerne pas tous les gènes. En effet, une partie des ARN produits à partir de la transcription d'un gène sont dits *non-codants* car ils ne sont pas traduits en protéines.

Différents mécanismes permettent la régulation de l'expression des gènes, et ce à différentes étapes de ce processus.

La régulation de l'expression peut intervenir au moment de la transcription de l'ADN en ARN messager. Les éléments régulateurs sont des petites séquences d'ADN qui permettent de moduler l'expression de certains gènes par le recrutement de protéines régulatrices (dénommées *activateurs* ou *répresseurs*) qui influent sur la stimulation d'un promoteur. Ces éléments peuvent se situer à des milliers de paires de bases des promoteurs. Bien que linéairement éloignés, c'est le repliement tridimensionnel de la chromatine qui permet une proximité entre un élément régulateur et un promoteur, impliquant ainsi la régulation de l'expression d'un gène. Ce mécanisme est illustré sur la figure 1.2. L'expression génique est donc particulièrement dépendante de la structure 3D du génome, en particulier de ce que l'on nomme l'accessibilité de la chromatine, c'est-à-dire son degré de compaction. On différencie ainsi l'euchromatine, forme ouverte de la chromatine, propice à la transcription, de l'hétérochromatine, forme plus condensée et où la transcription est plus rare.

Des éléments dits épigénétiques, c'est-à-dire ne relevant pas de la modification de la séquence d'ADN, peuvent aussi impacter l'expression génique. Par exemple, la méthylation de l'ADN, qui correspond à l'ajout d'un groupe méthyle (CH_3) sur certaines bases, peut entraîner la répression d'un gène si la méthylation a lieu dans sa région promotrice.

Ces exemples illustrent l'importance de l'organisation tridimensionnelle du génome pour la réalisation des fonctions cellulaires et, ainsi, le lien entre structure et fonction. Au

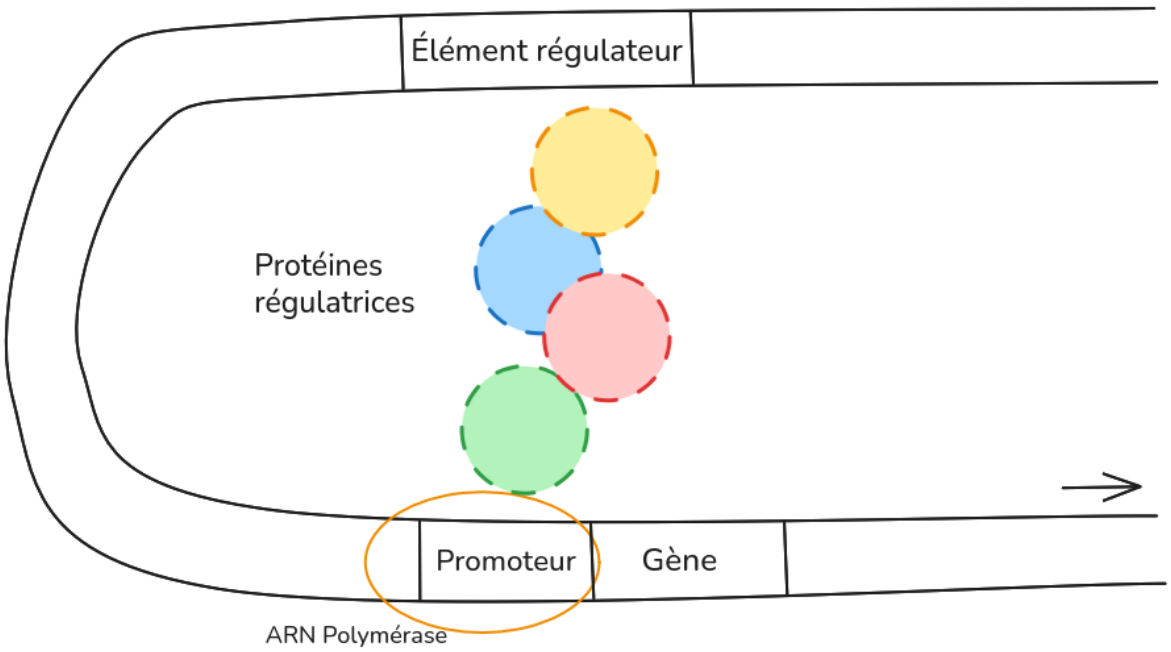


FIGURE 1.2 – **Illustration de la régulation de la transcription.** Une ARN polymérase se fixe au promoteur d'un gène afin d'initier la transcription. Le repliement de l'ADN permet une proximité spatiale entre un promoteur et un élément régulateur qui recrute des protéines régulatrices agissant sur la transcription. La flèche représente le sens de transcription.

cours du temps, il a été démontré que cette organisation est hiérarchique et multi-échelle avec des types de structures caractérisées en fonction de la résolution à laquelle le génome est observé.

1.2 Caractérisation de la structure 3D du génome et données Hi-C

Afin de pouvoir étudier plus finement le lien entre structure et fonction, il est nécessaire de cartographier la conformation 3D du génome. Pour ce faire, diverses méthodes existent. Des méthodes telles que FISH (Fluorescent In Situ Hybridization) [Kempfer and Pombo, 2020] permettent de localiser visuellement une séquence d'ADN ciblée par fluorescence et microscopie et d'estimer ainsi la proximité spatiale entre deux séquences d'intérêt. Cependant, ce type de méthode ne permet pas d'obtenir une cartographie globale de la conformation 3D d'un chromosome ou du génome.

Un second type de méthode, apparu sous l'acronyme 3C (Chromosome Conformation Capture) [Dekker et al., 2002] permet de mesurer, pour deux segments génomiques donnés, leur nombre moyen d'interactions et d'utiliser cette information comme indicateur de la distance spatiale entre ces deux segments. En effet, plus deux segments génomiques sont proches en trois dimensions, plus la probabilité qu'ils établissent une interaction durant

l'expérience est élevée. Une des méthodes les plus utilisées aujourd'hui pour caractériser la conformation 3D du génome – la méthode Hi-C (High-Throughput Conformation Capture) [Lieberman-Aiden et al., 2009, Belton et al., 2012] – est une extension du protocole 3C à l'étude simultanée d'un grand nombre de paires de segments. Ainsi, l'expérience Hi-C permet d'obtenir une cartographie globale de l'organisation tridimensionnelle du génome en caractérisant à la fois la proximité spatiale entre segments génomiques d'un même chromosome – dits *intra-chromosomiques* – mais également entre segments appartenant à des chromosomes différents – dits *inter-chromosomiques*.

1.2.1 Expérience Hi-C

On peut séparer l'Hi-C en deux catégories : *bulk* et *single-cell*. La première consiste à réaliser l'analyse pour un ensemble de cellules tandis que la seconde se concentre sur la caractérisation de la conformation du génome à l'échelle d'une cellule unique. Dans ce second cas – sur lequel on ne s'attardera pas ici – les comptages sont épars : l'interaction entre deux fragments de chromatine peut avoir été observée au plus une fois.

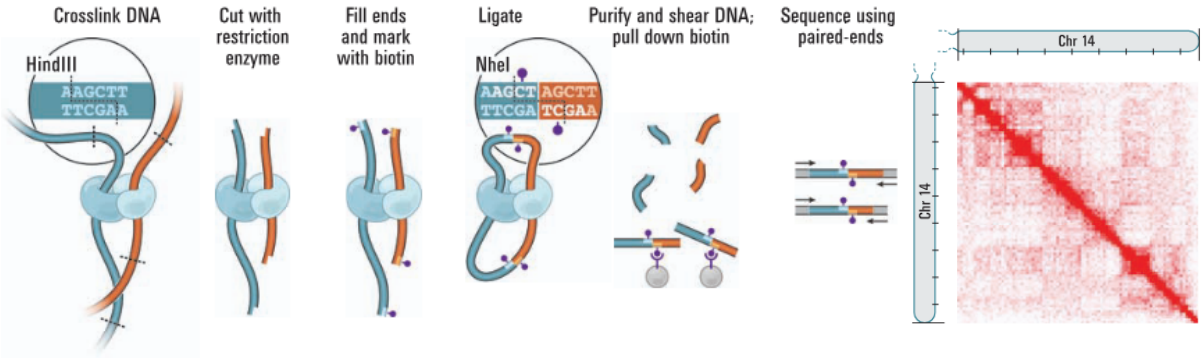


FIGURE 1.3 – Gauche : Représentation des différentes étapes de la technique Hi-C. Droite : matrice Hi-C obtenue pour le chromosome (humain) 14. Images issues de Lieberman-Aiden et al. [2009].

La première étape de l'expérience Hi-C consiste en la fixation des contacts grâce au formaldéhyde. Cette étape permet de s'assurer que les proximités entre positions génomiques seront préservées au cours des prochaines étapes de l'expérience. La seconde étape, observable sur la figure 1.3, consiste en la fragmentation de la chromatine par une enzyme de restriction qui reconnaît des séquences spécifiques de nucléotides et sectionne la chromatine à ces endroits (dits sites de restriction). Ensuite, les extrémités des fragments découpés sont marqués par de la biotine. S'opère alors la ligature, permettant aux fragments qui étaient proches en 3D lors de la première étape de fixation d'être associés. On obtient ainsi des fragments hybrides, composés de régions génomiques proches spatialement. La purification de l'ADN consiste ensuite à séparer les fragments qui ne sont pas issus d'une ligature de ceux qui ont été ligaturés et à ne garder que ces derniers. L'ensemble des fragments issus de l'expérience est alors dénommé « librairie Hi-C ». La dernière étape, représentée sur la figure 1.4, consiste au séquençage et à l'alignement des séquences de la librairie Hi-C. Lors du séquençage, chaque fragment est « lu » par ses deux extrémités afin d'obtenir deux séquences. L'alignement de ces deux séquences sur

un génome de référence permet alors de déterminer la position génomique des régions composant le fragment hybride.

Le résultat de l'expérience peut être stocké dans une matrice appelée « matrice Hi-C ». Comme représenté sur la figure 1.3, à droite, si l'expérience Hi-C est réalisée à l'échelle d'un chromosome (ici le chromosome 14), celui-ci est découpé en intervalles génomiques de taille régulière appelés « bins ». La taille de ces intervalles génomiques est la résolution. Ainsi, pour deux bins i et j, le pixel (i, j) de la matrice est utilisé pour stocker le nombre d'interactions entre ces deux bins, c'est-à-dire le nombre de paires de lectures faites durant l'expérience et correspondant à une ligature entre un segment appartenant au bin i et un segment appartenant au bin j. Le nombre d'interactions enregistrées pour un pixel (i, j) au cours de l'expérience est alors un proxy de la proximité spatiale entre les positions génomiques i et j. Précisons que les matrices Hi-C sont symétriques et qu'on retrouve donc souvent des représentations limitées à la demi-matrice.

En relation avec l'obtention des données Hi-C, on parlera souvent de la « profondeur de séquençage » qui détermine la couverture (nombre de lectures) s'alignant sur un nucléotide. Ainsi, plus cette profondeur est élevée, plus le nombre total de comptages dans la matrice est élevé. Cependant, il existe un compromis entre la profondeur de séquençage et la résolution. La résolution choisie (donc la taille des bins) dépend de plusieurs facteurs, dont l'enzyme de restriction utilisée durant l'expérience Hi-C. En effet, comme l'enzyme découpe l'ADN à un endroit où elle reconnaît un motif, si celle-ci est capable de reconnaître un motif qui apparaît fréquemment, elle peut produire des segments de plus petite taille et ainsi produire des matrices plus « fines » avec des bins de petite taille. Or, lorsqu'on diminue la taille des bins, le nombre de lectures s'alignant sur un bin est alors plus faible.

Pouvoir regarder ces données à une résolution élevée (avec des bins de petite taille) permet d'étudier des structures fines représentées sur la matrice Hi-C tandis qu'avoir une profondeur de séquençage élevée permet d'effectuer des analyses – en particulier statistiques – avec une plus grande confiance en les résultats.

1.2.2 Pré-traitements et normalisations des données Hi-C

De nombreux biais, qu'ils soient biologiques ou techniques, peuvent compromettre la qualité des données Hi-C. Afin de les prendre en compte et de les corriger, il existe plusieurs pré-traitements tels que les filtrages d'interactions et la normalisation des matrices [Forcato et al., 2017, Lajoie et al., 2015, Pal et al., 2019]. Le choix de ces pré-traitements n'est pas universel et dépend du pipeline utilisé pour le traitement et l'analyse des données Hi-C. Comme représenté sur la figure 1.4, à l'issue de l'expérience Hi-C, plusieurs étapes sont nécessaires pour obtenir les matrices à partir des lectures. Après l'étape d'alignement, on peut par exemple chercher à corriger des erreurs en éliminant les lectures dont la distance au site de restriction le plus proche semble aberrante. Aussi, après avoir assigné les paires de lectures aux bins, on peut éliminer les bins dont la couverture semble faible comparée à la distribution totale des comptages. Une fois les filtrages effectués, l'étape de normalisation est typiquement appliquée.

Il existe deux types d'approches de normalisation de matrices Hi-C : explicites et im-

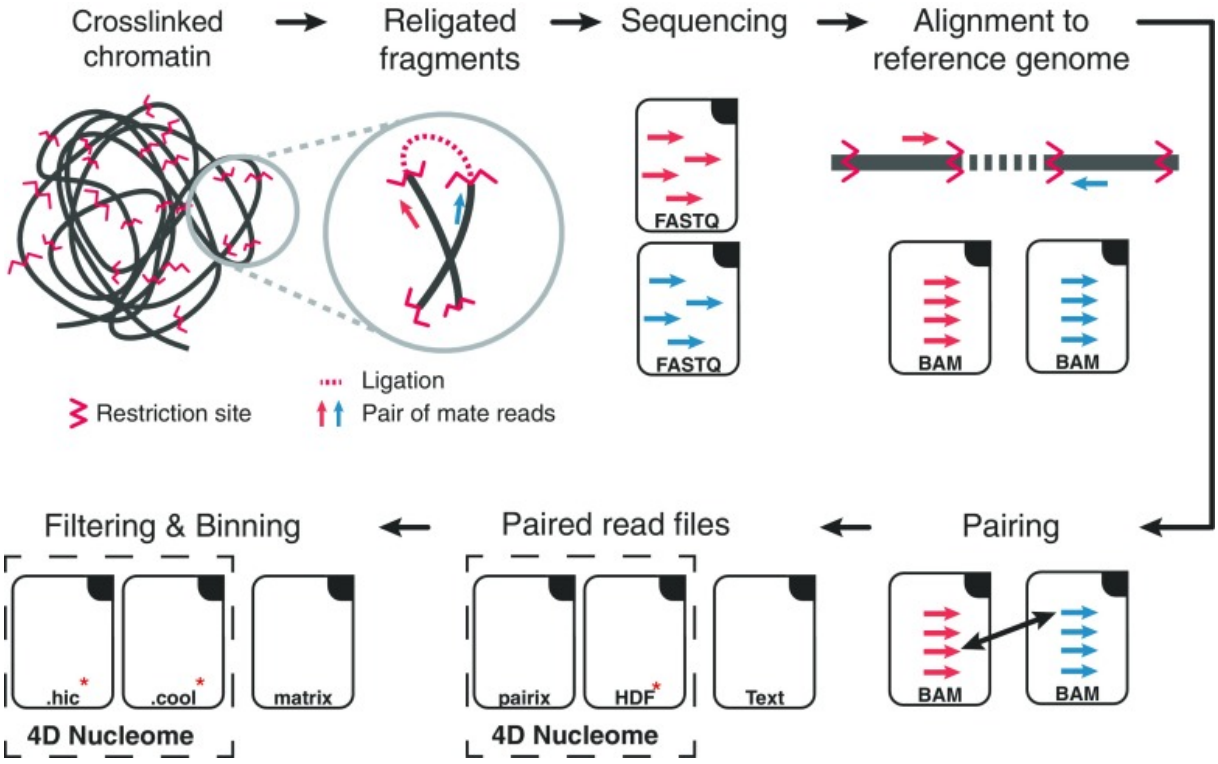


FIGURE 1.4 – Représentation des étapes de l'expérience Hi-C permettant de passer du séquençage des lectures, à leur alignement, leur appariement puis les filtrages et l'assignation des lectures aux bins avant, enfin, la création de la matrice Hi-C. Figure issue de Pal et al. [2019].

plicites. Les approches explicites consistent à quantifier des biais comme la taille des fragments, le GC pourcent (ratio de nucléotides G et C dans un segment), la « mappabilité » (unicité de l'alignement d'un fragment à une séquence, qui dépend de la taille du fragment et de la répétitivité de la séquence dans le génome) puis à les modéliser en utilisant des outils probabilistes ou statistiques [Hu et al., 2012, Yaffe and Tanay, 2011] avant d'appliquer une correction aux comptages pour chaque bin. Les approches implicites consistent à corriger les comptages afin de faire en sorte que chaque bin soit représenté de la même manière, c'est-à-dire que la somme des comptages associés à un bin soit la même pour tous. De manière non exhaustive, ce type de normalisation peut être réalisé par les méthodes ICE [Imakaev et al., 2012], SCN [Cournac et al., 2012] ou KR (Knight-Ruiz) [Knight and Ruiz, 2013].

Les normalisations mentionnées précédemment s'appliquent individuellement sur chaque matrice et ont pour but de rendre comparables les comptages de bins différents. Cependant, il existe des normalisations qui s'opèrent sur plusieurs matrices à la fois, notamment quand l'objectif est de comparer entre elles plusieurs matrices Hi-C. En outre, en fonction du type d'analyse réalisée sur les données Hi-C, telle que l'identification de structures tridimensionnelles, il est nécessaire d'appliquer des normalisations spécifiques.

1.2.3 Structures tridimensionnelles hiérarchiques

À partir des données Hi-C, il est possible d'identifier un certain nombre de structures tridimensionnelles. Les plus communes sont les territoires chromosomiques, les compartiments A/B, les TADs et les boucles de chromatine. Une représentation de ces différentes structures hiérarchiques peut être observée sur la figure 1.5. Ces structures se situent à des niveaux d'échelle très différents dans le génome et leur identification requiert des matrices Hi-C à des résolutions appropriées. Dans la suite, je présente de manière non exhaustive quelques-unes de ces structures et les outils permettant de les identifier à partir des données Hi-C.

Territoires chromosomiques

À l'échelle la plus grande, les chromosomes sont organisés au sein du noyau de la cellule en territoires chromosomiques. L'organisation des chromosomes en territoires distincts est bien connue, et ce depuis avant même l'apparition des techniques de profilage telles que l'Hi-C [Cremer and Cremer, 2001, Meaburn and Misteli, 2007]. En particulier, le lien entre cette organisation en territoires et l'expression génique est bien connu [Cremer et al., 1993]. Il est possible d'inférer la structure des territoires chromosomiques à partir de l'expérience Hi-C [Das et al., 2020]. La figure 1.6 représente une matrice Hi-C à l'échelle de trois chromosomes, avec les interactions intra-chromosomiques (*cis*) le long de la diagonale et les interactions inter-chromosomiques (*trans*) en dehors. On peut observer que les interactions *trans* sont bien moins nombreuses que les *cis*, mais elles ne sont pas toutes nulles !

Compartiments A/B

Comme cela peut être observé sur la figure 1.5, à l'échelle du chromosome, celui-ci est divisé en un compartiment A dit actif et un compartiment B dit inactif. Ces deux compartiments correspondent respectivement à l'euchromatine et l'hétérochromatine mentionnées dans la section 1.1. Il apparaît que l'euchromatine se situe plus au centre du noyau tandis que l'hétérochromatine est à la périphérie [Hildebrand and Dekker, 2020]. Par ailleurs, l'euchromatine semble être plus « active » que l'hétérochromatine au niveau transcriptionnel et donc contenir des gènes plus exprimés.

On peut observer les deux compartiments d'un chromosome et une portion de la matrice Hi-C correspondante sur la figure 1.6. Le chromosome étant représenté linéairement sur les deux axes de la matrice Hi-C, on a une alternance de compartiments A et B. La détection de compartiments à partir de données Hi-C s'effectue généralement en deux étapes : une étape de normalisation et une étape d'analyse en composantes principales. La normalisation la plus largement utilisée dans ce cadre est la normalisation « *observed/expected* » qui consiste à corriger l'influence de la distance à la diagonale sur les comptages. En effet, comme noté dans Lieberman-Aiden et al. [2009], dans les données Hi-C, la relation entre la probabilité d'interaction entre deux bins i et j et leur distance sur le chromosome $|i-j|$ suit une loi de puissance. Ainsi, on corrige le comptage associé au couple de bins (i, j) en le divisant par le nombre moyen d'interactions enregistrées pour les paires de bins situées sur la même diagonale. Une matrice Hi-C ainsi normalisée

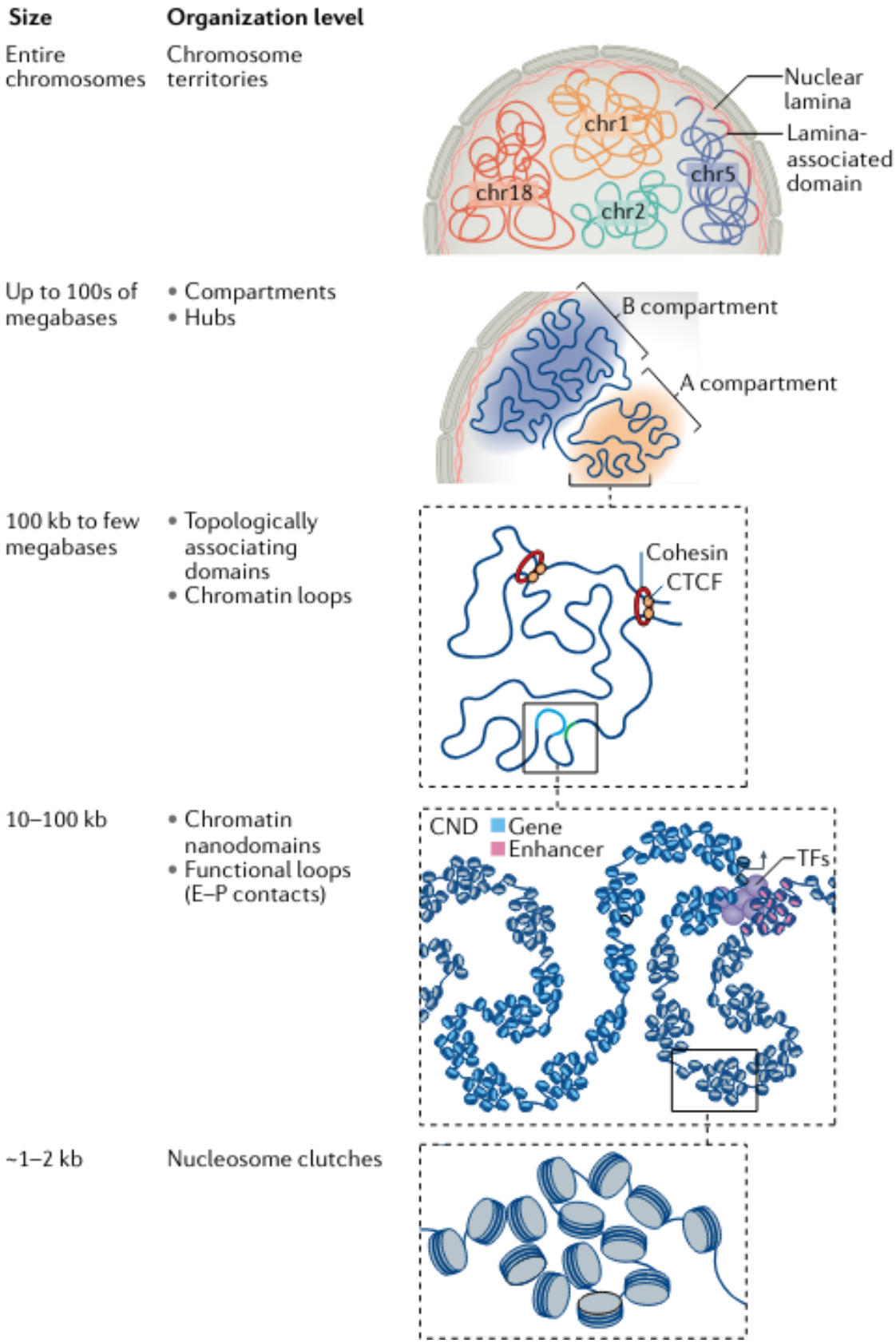


FIGURE 1.5 – Représentation de l'organisation 3D du génome et des structures identifiées à différents niveaux d'échelle. Figure extraite de Jerković and Cavalli [2021].

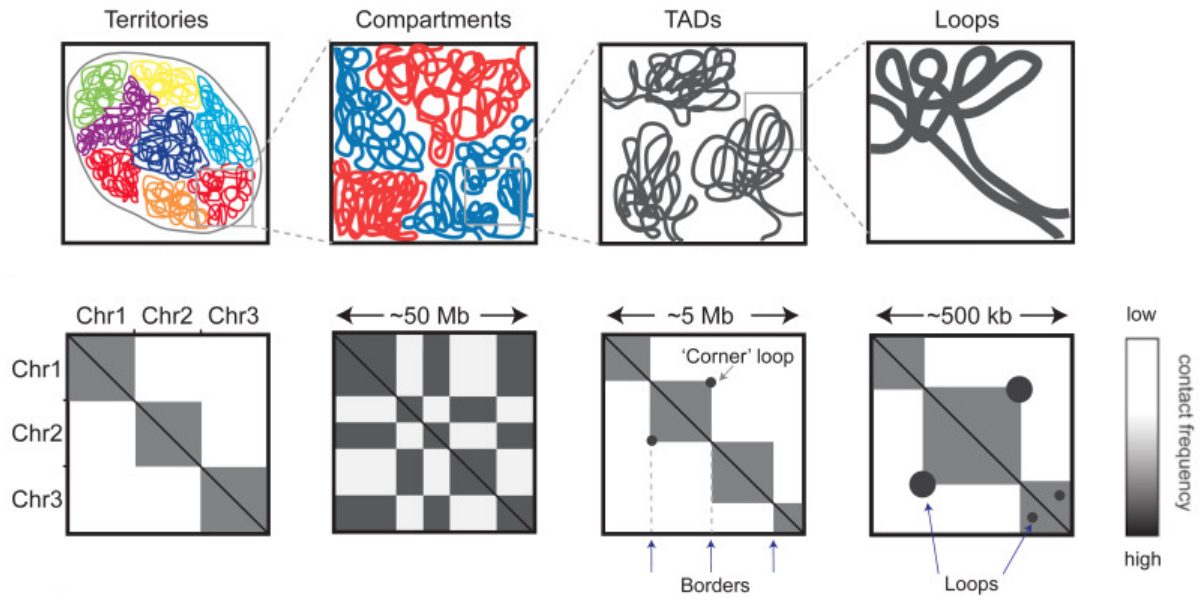


FIGURE 1.6 – Représentation des différentes structures tridimensionnelles observées dans le génome et leur aspect au sein d'une matrice Hi-C. Figure adaptée de Sikorska and Sexton [2020].

a alors un aspect en « damiers », comme on peut l'observer sur la figure 1.6.

La seconde étape consiste à réaliser une analyse en composantes principales (ACP) [Lajoie et al., 2015], permettant de retrouver deux vecteurs propres principaux de signes opposés, indiquant l'appartenance de chaque bin au compartiment A ou B. Il est aussi possible de détecter les compartiments en utilisant l'algorithme de clustering des k-means [Maigné et al., 2025, Wagstaff et al., 2001]. Ces méthodes permettent simplement de diviser le chromosome en deux compartiments mais l'assignation au compartiment A ou B doit se faire en utilisant des informations biologiques supplémentaires.

TADs

En regardant le chromosome à une résolution d'environ 1 Mb, on peut observer des régions génomiques appelées TADs (*Topologically Associating Domains*). Un TAD correspond à un segment chromosomique au sein duquel les positions ont des interactions fréquentes entre elles et des interactions plus rares avec des positions en dehors du segment [Dixon et al., 2012]. Ainsi, comme on peut l'observer sur la figure 1.6, un TAD correspond à un triangle le long de la diagonale d'une matrice Hi-C caractérisé par ses deux frontières. On peut également voir que certains TADs ont un « point » au sommet qui correspond à la boucle formée entre les deux frontières. Cette boucle se forme au cours d'un processus nommé « extrusion » [Fudenberg et al., 2016] rendu possible par l'action des protéines CTCF et cohésine, représenté sur la figure 1.7. De plus, il a été montré que ces deux protéines étaient fréquemment toutes deux localisées aux frontières des TADs [Dixon et al., 2012, Rao et al., 2014]. D'un point de vue fonctionnel, les TADs sont particulièrement impliqués dans la régulation de l'expression des gènes, comme discuté dans

la section 1.2.4.

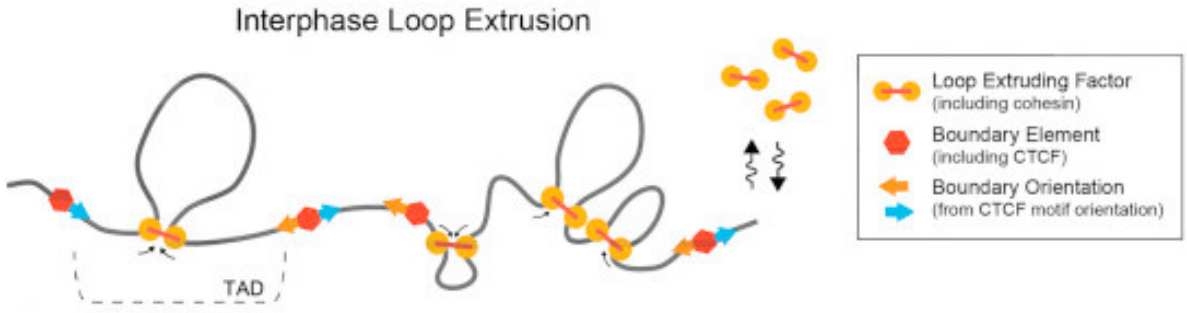


FIGURE 1.7 – Représentation du processus d'extrusion de boucle. Figure adaptée de Nagasaka et al. [2023].

De nombreuses méthodes, étudiées dans plusieurs benchmarks [Dali and Blanchette, 2017, Sefer, 2022, Xu et al., 2024], proposent d'identifier des TADs à partir de données Hi-C. Concernant la normalisation des matrices, il n'y a pas de standard, ce sont souvent des normalisations intra-matrices de type ICE ou KR (Knight-Ruiz) qui sont appliquées. En termes de prédiction de TADs, certaines approches consistent à produire une métrique 1D sur la matrice le long du chromosome, puis à identifier des changements dans cette métrique correspondant à des frontières [Lyu et al., 2020]. D'autres méthodes s'appuient sur la représentation des matrices Hi-C par des graphes, avec des sommets correspondant aux bins (positions génomiques) et des arêtes pondérées relativement au nombre de comptages enregistrés entre chaque paire de bins [Gong et al., 2023]. Il existe des approches basées sur du clustering à partir des bins ou des pixels, permettant parfois d'avoir une approche hiérarchique dans la détection des TADs [Soler-Vila et al., 2020, Ambroise et al., 2019]. Notons que certaines méthodes s'appuient sur une modélisation statistique, en supposant par exemple une distribution sur les frontières de TADs et en cherchant à faire de la détection de ruptures [Lévy-Leduc et al., 2014]. Ces exemples ne sont pas exhaustifs et illustrent simplement différentes stratégies de prédicteurs de TADs.

De la comparaison des prédicteurs de TADs, il ressort qu'il n'existe pas de définition unique du TAD à travers les différentes approches. Par ailleurs, il n'existe pas non plus de jeu de données sur lequel on connaisse les TADs *a priori*. Dans le cadre de l'évaluation de méthodes de prédictions, les performances sont généralement évaluées sur des jeux de données où les TADs ont été définis manuellement à partir de l'observation de la carte Hi-C, introduisant des biais dans la comparaison. De plus, les TADs sont considérés par certains comme des structures hiérarchiques imbriquées avec des *nested* et des *sub* TADs. La capacité à détecter ces différents niveaux d'échelle de TADs dépend alors grandement des hypothèses formulées dans chacune des approches de prédicteurs et de la résolution disponible pour les données Hi-C. Comme souligné dans Zufferey et al. [2018], la grande variabilité dans les détections (en terme de nombre et taille de TADs) en fonction de la résolution des matrices sur une même méthode met en avant l'intérêt d'utiliser une approche modélisant les TADs de manière hiérarchique. Dans ce cadre, l'évaluation et la comparaison des prédicteurs de TADs est délicate et il est souvent recommandé d'utiliser plusieurs prédicteurs.

Boucles de chromatine

Comme mentionné dans le paragraphe précédent, la structure permettant de mettre en proximité 3D des éléments qui se trouvent linéairement éloignés sur le chromosome est la boucle de chromatine [Rao et al., 2014]. Formée au cours d'un processus d'extrusion, la boucle est vue comme un point sur la carte Hi-C, observable sur la figure 1.6. Identifier des boucles requiert généralement des données à haute résolution, celles-ci correspondant à un contact ponctuel entre deux positions génomiques. La boucle est un élément fondamental dans l'organisation spatiale du génome puisqu'elle permet, entre autres, de mettre en contact des gènes et leurs éléments régulateurs, possiblement (très) éloignés le long du chromosome.

Il existe un grand nombre d'outils de détection de boucles et des comparaisons sont proposées dans Liu et al. [2023], Chowdhury et al. [2024]. Globalement, il en ressort qu'il existe une grande variabilité dans les détections faites par les différentes méthodes et que le nombre de boucles détectées décroît lorsque la taille des bins augmente. Par ailleurs, certaines méthodes ne peuvent pas être appliquées à des données dépassant les 10 ou 25kb de résolution. Comme pour la détection de TADs, il n'existe pas de jeu de données de référence dans lequel les boucles seraient connues, la validation des résultats se fait donc avec l'étude de marqueurs biologiques comme la présence de la cohésine dans les boucles identifiées.

Identification de structures multi-échelles

Comme évoqué précédemment, l'organisation tridimensionnelle du génome est multi-échelle et, de ce fait, on pourrait vouloir identifier plusieurs niveaux de structures avec une même méthode plutôt que de devoir utiliser un outil pour chaque type de structure ciblée. Une telle approche est proposée dans Kumar et al. [2020] à travers un clustering hiérarchique de matrices Hi-C sous forme de segmentation, s'affranchissant d'une contrainte de contiguïté sur les clusters. Cette approche permet, en fonction de la résolution des données, de retrouver à la fois des clusters correspondant à des compartiments et des clusters correspondant à des TADs.

Comme on a pu le voir au travers des différentes méthodes de prédiction de structures, la plupart de celles que je mentionne ici n'ont pas de définition consensuelle. Pour valider l'existence de ces structures et comprendre les éléments et mécanismes biologiques responsables de leur apparition et de leur maintien, il est d'usage de chercher à les perturber expérimentalement pour voir l'impact fonctionnel de cette perturbation.

1.2.4 Perturbations de structures et rôle fonctionnel

Il a été montré que certaines structures tridimensionnelles telles que les TADs étaient fortement conservées entre des types cellulaires différents et des espèces comme l'homme et la souris [Dixon et al., 2012]. Cependant, malgré la conservation de celles-ci, l'architecture 3D du génome n'est pas universelle et une grande variabilité subsiste dans cette organisation, même au sein d'un organisme. Par exemple, des changements dans la conformation

du génome sont observés lors du processus de différentiation cellulaire comme lors de la spécialisation des cellules neuronales [Bonev et al., 2017]. Par ailleurs, afin d'identifier les éléments responsables du maintien ou de la création d'une structure tridimensionnelle, on peut chercher à la perturber. Aussi, perturber une structure et observer les impacts provoqués par ce changement permet d'identifier des fonctions qui lui sont associées.

Pour élucider le rôle de la cohésine dans la formation de boucles de chromatine, Rao et al. [2017] ont réalisé une étude consistant à dépléter la cohésine puis à la réintroduire dans des cellules cancéreuses humaines. L'observation des matrices Hi-C avant et après déplétion de la cohésine montre la disparition des « points » caractérisant les boucles de chromatine. Par ailleurs, cette déplétion n'a pas impacté les compartiments des matrices, suggérant la responsabilité de la cohésine dans la formation de boucles de chromatine mais pas dans celle des compartiments. De plus, cette déplétion semble peu, voire pas du tout, impacter la régulation de l'expression des gènes.

Une étude de malformations congénitales affectant le développement des membres est présentée dans Lupiáñez et al. [2015]. Ces malformations sont causées par des variations de la séquence d'ADN impliquant la délétion, l'inversion ou la duplication d'un segment contenant un ou plusieurs gènes. Ces modifications de la séquence entraînent des changements dans les TADs avoisinants, comme la suppression ou le décalage d'une frontière de TAD. Comme on peut l'observer sur la figure 1.8, la régulation de l'expression des gènes est impactée par cette réorganisation. En effet, l'inversion d'une séquence peut par exemple placer un élément régulateur de l'autre côté d'une frontière, dans un TAD différent. Cet élément régulateur peut alors moduler l'expression d'un gène sur lequel il n'aurait eu auparavant aucun effet. On observe des conséquences similaires sur la régulation de l'expression des gènes pour les duplications ou délétions de séquences. Cette étude permet d'illustrer le rôle fonctionnel de la structure tridimensionnelle du génome à travers le lien établi entre des perturbations de la structure des TADs et la modulation de l'expression des gènes.

Le premier exemple que j'ai présenté – relatif au rôle de la cohésine – illustre la déplétion d'un élément et l'observation des dégradations de structures qui en résultent, mettant en avant son rôle dans leur formation. Le second exemple permet quant à lui d'illustrer le lien entre structures tridimensionnelles et processus biologiques. Pour ces deux exemples, des données Hi-C ont été produites dans deux conditions biologiques différentes et c'est leur comparaison au travers de l'analyse différentielle qui a permis d'obtenir des résultats.

1.3 Analyse différentielle de données Hi-C

Comme expliqué dans la section 1.2.4, les modifications de la structure tridimensionnelle entre deux conditions biologiques peuvent avoir des conséquences importantes et il est donc primordial de pouvoir les détecter afin de comprendre les mécanismes qu'elles impactent.

L'analyse différentielle de données Hi-C consiste à comparer des matrices obtenues dans deux conditions biologiques d'intérêt dans l'objectif d'identifier des régions génomiques spatialement réarrangées. Dans la suite, j'introduis le cadre mathématique de l'analyse

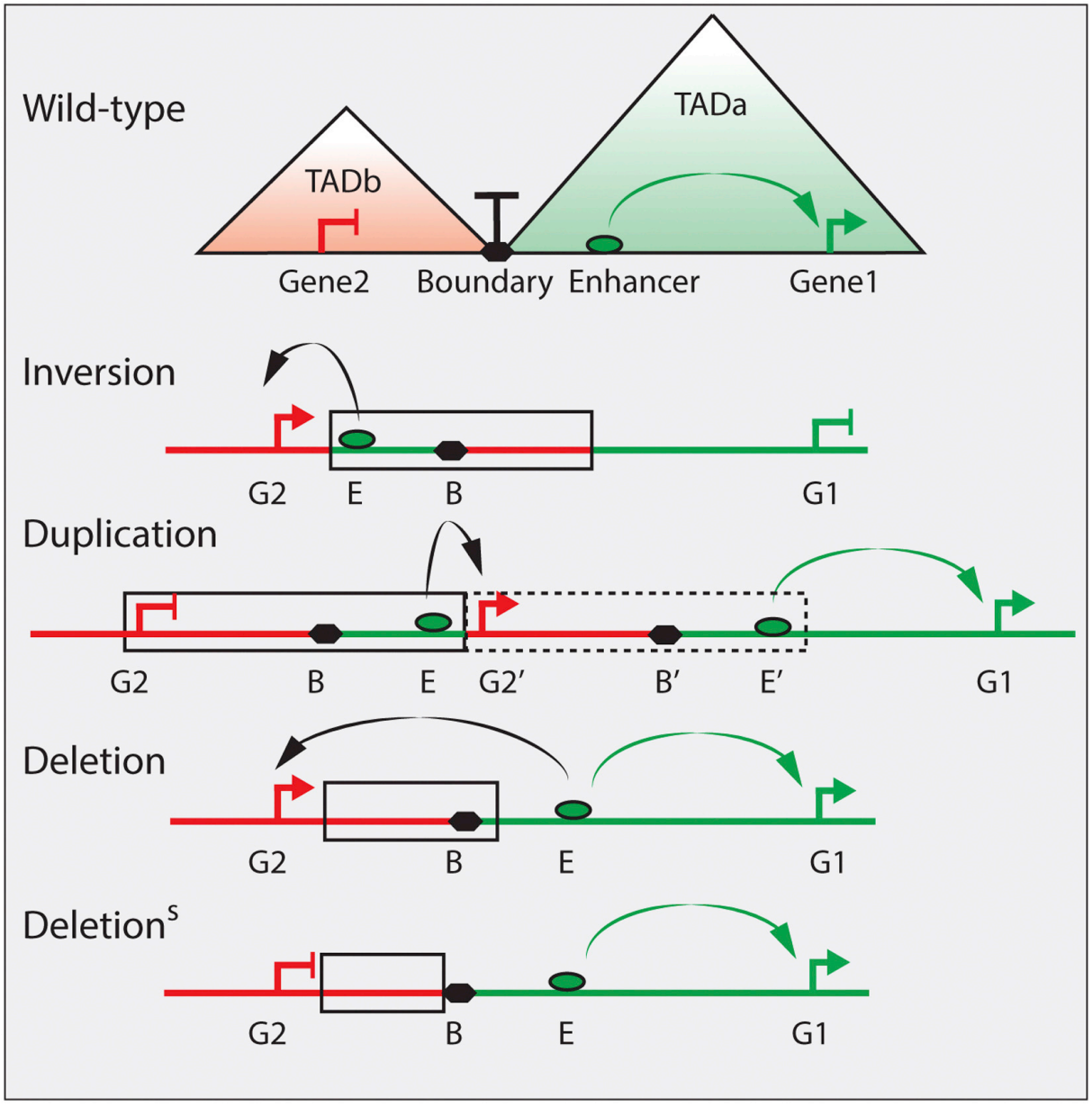


FIGURE 1.8 – **Illustration du lien entre modifications de séquence, perturbations de frontières de TADs et régulation de l'expression génique.** Représentation de deux TADs (TADa et TADb) séparés par une frontière. Chaque TAD contient un gène, ces deux gènes sont régulés indépendamment (l'élément régulateur situé dans le TADa régule le gène n°1 uniquement). L'inversion de la séquence contenant la frontière de TAD et l'élément régulateur implique le déplacement de l'élément régulateur du côté de la frontière où se trouve le gène n°2 et une modulation de son expression. La duplication de la séquence contenant : le gène n°2, la frontière de TAD et l'élément régulateur, implique la régulation (anormale) de la copie du gène n°2 par l'élément régulateur. De même, si la déplétion intervient sur une séquence contenant la frontière, celle-ci disparaît et n'empêche plus la modulation du gène n°2 par l'élément régulateur. Figure extraite de Lupiáñez et al. [2015].

différentielle de données Hi-C puis je présente les pré-traitements nécessaires à la comparaison de matrices Hi-C ainsi que les deux principales approches d'analyse différentielle : l'analyse différentielle basée sur les pixels et l'analyse différentielle basée sur les structures.

1.3.1 Définition mathématique

Formellement, on se place dans le cadre de la comparaison de deux conditions biologiques d'intérêt $\mathcal{C}_1$ et $\mathcal{C}_2$. On considère un ensemble de $r = r_1 + r_2$ matrices Hi-C symétriques de taille $d \times d$. On note $M^{g,l}$ la matrice correspondant au l-ème réplicat de la condition $\mathcal{C}_g$ où $l \in \{1, \cdots, r_g\}$ et $g \in \{1, 2\}$. De plus, chacune de ses entrées $m_{ij}^{g,l}$ correspond au nombre d'interactions enregistrées durant l'expérience Hi-C entre les « bins » (intervalles génomiques) i et j. La résolution de la matrice correspond à la taille (fixée) d'un bin en paire de bases. L'objectif de l'analyse différentielle de données Hi-C est d'identifier des conformations génomiques différentielles entre les conditions $\mathcal{C}_1$ et $\mathcal{C}_2$. Pour ce faire, il existe deux types principaux de méthodes : l'analyse différentielle basée sur les pixels et l'analyse différentielle basée sur les structures. Dans la suite, je présente ces deux types de méthodes, les normalisations qu'elles nécessitent, leur fonctionnement ainsi que leurs limites.

1.3.2 Analyse différentielle basée sur les pixels

L'analyse différentielle basée sur les pixels consiste à tester, pour chaque pixel (i, j), l'hypothèse : « L'interaction entre les deux positions génomiques i et j n'est pas différente entre les deux conditions biologiques $\mathcal{C}_1$ et $\mathcal{C}_2$ ».

Normalisation. Avant de réaliser une telle analyse, il est nécessaire de préparer les matrices Hi-C de manière adéquate. L'étape de filtrage, visant à enlever des bins (ou des pixels) pour lesquels les comptages seraient trop faibles ou sur lesquels nous aurions peu de confiance, n'est pas obligatoire ici car ils ne constituent pas un biais à proprement parler dans l'analyse comparative de ces données. Cependant, avoir un nombre de pixels plus faible à considérer peut permettre de réaliser moins de tests et ainsi gagner en puissance statistique [Bourgon et al., 2010].

Concernant la normalisation, on peut réaliser une normalisation intra-matrice mais ce n'est pas primordial dans ce cadre car on ne cherche pas à comparer des pixels entre eux au sein d'une même matrice mais plutôt des matrices entre elles, en particulier un même pixel dans différentes matrices. Ainsi, il est très important de réaliser une normalisation inter-matrices, les rendant comparables entre elles. Ce type de normalisation cherche à corriger des biais liés par exemple à des différences de profondeur de séquençage entre librairies.

Dans la normalisation MA [Dudoit et al., 2002], pour un pixel donné (i, j), on regarde le comptage associé dans deux matrices distinctes. On calcule ensuite (en échelle log) la différence entre ces comptages « M » et la moyenne des comptages « A ». En représentant, pour une paire de matrices donnée, les valeurs MA pour chaque pixel, on peut alors

observer s'il existe une tendance à trouver plus de différences pour les pixels ayant des comptages élevés. La normalisation LOESS [Cleveland, 1979] permet de corriger cette tendance en la modélisant par une régression locale puis en transformant les comptages pour ramener la tendance à 0. Deux exemples de MA-plots et des tendances associées sont représentés sur la figure 1.9.

Comme on l'a déjà noté, il existe un lien entre le nombre de comptages enregistrés pour un pixel et la distance à la diagonale. La normalisation MD [Stansfield et al., 2018] fonctionne de manière analogue à la MA en corrigeant la tendance entre la différence entre les comptages « M » et la distance à la diagonale « D ».

Dans le cadre de l'analyse différentielle de données Hi-C, on possède généralement des duplicats voire des réplicats dans chaque condition biologique. La normalisation doit permettre de rendre comparables toutes les matrices entre elles. Typiquement, une normalisation de type LOESS est alors appliquée de manière cyclique sur toutes les paires de matrices, indifféremment de la condition à laquelle elles appartiennent, jusqu'à ce que l'algorithme converge et que la tendance soit corrigée pour toutes les matrices. L'impact de la correction LOESS sur les MA-plots est représenté sur la figure 1.9.

Méthodes pour l'analyse. Il existe un certain nombre de méthodes différentes proposant de réaliser l'analyse différentielle de données Hi-C basée sur les pixels. Dans le chapitre 2, je présente un travail dans lequel nous avons proposé de décrire et d'évaluer ces méthodes. Ces différentes approches résultent en l'obtention d'une p-valeur (et d'une p-valeur ajustée) pour chaque pixel (i, j) des données. Cette approche d'analyse différentielle fournit des résultats locaux, avec des pixels différentiels potentiellement éparpillés dans la carte Hi-C et une interprétabilité limitée. Par ailleurs, le fait de réaliser une multitude de tests en simultané nous place dans le cadre particulier des tests multiples. J'aborde ces différents points dans la section 1.4.

Notons que la puissance de détection de ce type de méthode est fortement dépendante de la résolution des données. En effet, comme cela a été abordé plus tôt, les matrices issues de résolutions plus élevées ont tendance à avoir une plus faible couverture des bins donc un signal moins fort. Ce faible signal peut alors ne pas être détecté par les tests statistiques.

1.3.3 Analyse différentielle basée sur les structures

Comme mentionné dans la section 1.2.3, un certain nombre de structures génomiques tridimensionnelles ont été identifiées et validées par des expériences biologiques. On peut chercher à identifier des changements dans la conformation 3D du génome entre deux conditions biologiques en s'intéressant aux structures et en évaluant si elles ont subi des remaniements.

Que ce soit pour les compartiments [Chakraborty et al., 2022, Kai et al., 2023, Maigné et al., 2025], les TADs [Chen et al., 2018, Cresswell and Dozmorov, 2020, Mourad, 2022, Hua et al., 2024, Liu and Ma, 2024] ou encore les boucles [Durand et al., 2016], l'analyse différentielle de ces structures repose généralement sur leur prédiction dans chacune

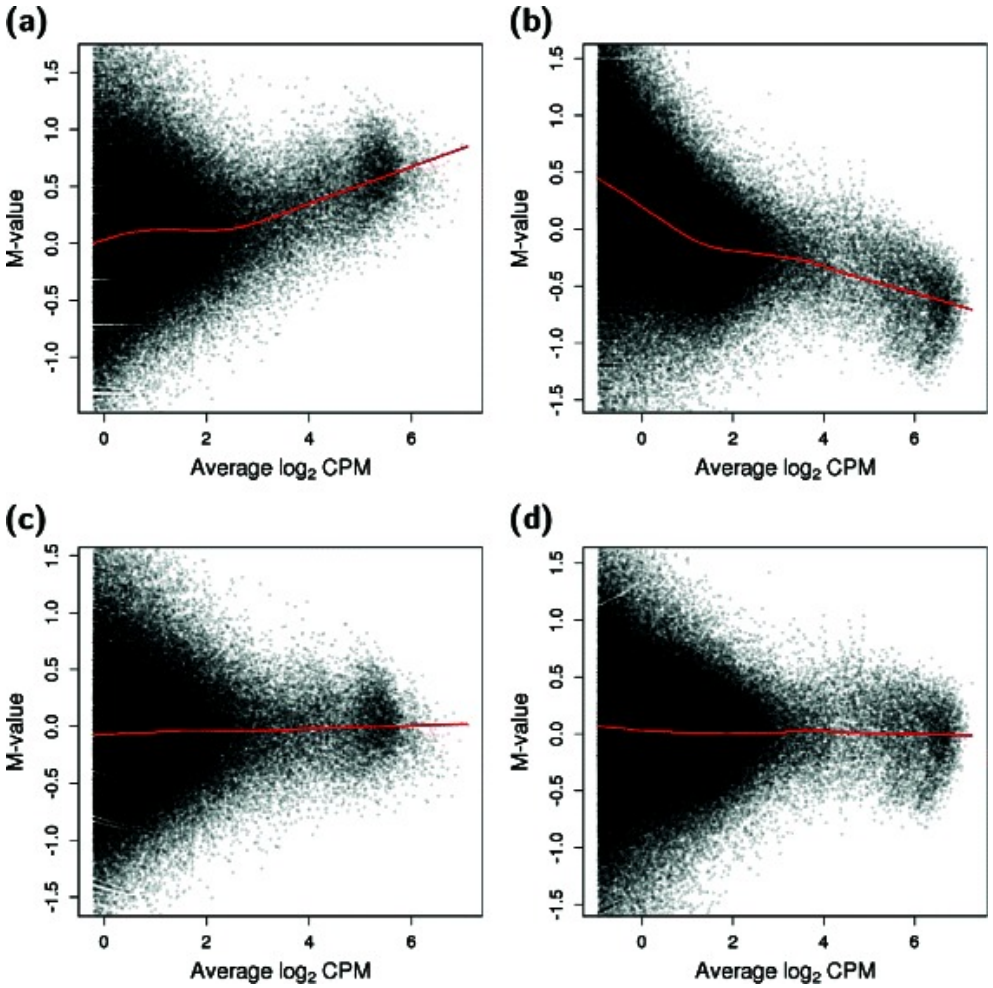


FIGURE 1.9 – **MA-plots avant et après correction.** Sur les quatre schémas sont représentés le M (log-fold change des comptages entre deux réplicats, corrigé par la taille de la librairie) par rapport au A (nombre moyen de comptages par million). Les plots **(a)** et **(b)** représentent deux jeux de données différents. Les plots **(c)** et **(d)** sont, respectivement, les versions corrigées des plots **(a)** et **(b)**. La tendance LOESS est représentée en rouge. Figure issue de Lun and Smyth [2015].

des conditions étudiées puis leur comparaison à travers, par exemple, une modélisation statistique. Cependant, ce type d'approche résulte en des prédictions différentielles peu comparables entre méthodes pour un même type de structure. En effet, il n'existe pas de définition universelle pour chacune des structures tridimensionnelles regardées. Par exemple, il n'y a pas de consensus sur la taille d'un TAD et celui-ci peut être vu comme une structure hiérarchique par certaines approches.

Enfin, notons que ces méthodes sont souvent limitées du fait de leur hypothèse principale : elles supposent que la région différentielle entre deux conditions est elle-même une structure définie, un TAD ou un compartiment par exemple. Or, les modifications subies par une structure peuvent être d'origine et de forme très différentes et engendrer une région différentielle qui ne correspond pas à la définition donnée d'une structure. Par ailleurs, on peut également avoir des changements quantitatifs, une compartimentation

moins forte ou des interactions plus faibles au sein d'un TAD par exemple. Ce type de changement est pris en compte par certains outils seulement.

1.4 Tests statistiques

Dans la section suivante, je présente les motivations de la réalisation d'un test statistique. J'aborde rapidement la construction d'un test et ce qui en découle : la p-valeur. J'introduis ensuite les deux types d'erreurs que l'on peut faire quand on prend la décision de rejeter (ou non) une hypothèse à l'issue d'un test. Enfin, j'introduis la problématique liée à la réalisation de tests multiples ainsi que l'interprétation de ces résultats dans le cadre de l'analyse différentielle de données Hi-C.

Pour rédiger cette section, je me suis appuyée sur Tassi [2004].

1.4.1 Définition et construction d'un test statistique

L'objectif d'un test statistique est de fournir une règle de décision relativement au rejet d'une hypothèse H_0 émise sur une population. On possède des observations issues de cette population : $x = (x_1, \cdots, x_n)$.

En fonction de H_0 et des hypothèses faites sur les observations (relativement à leur distribution par exemple), on peut construire un test à partir d'une statistique T, fonction des observations et dont la loi sous H_0 – c'est-à-dire, si l'hypothèse H_0 est vraie – est connue. Pour la plupart des hypothèses H_0 que l'on pourra poser, des tests existent déjà. Dans Kanji [2006], une centaine de tests statistiques différents sont présentés !

Une fois T obtenue, on peut, en la comparant à la distribution qu'elle devrait avoir sous H_0, produire une probabilité critique p, également appelée p-valeur. La p-valeur indique la plus petite valeur de α telle que H_0 soit rejetée où $\alpha = \mathbb{P}_{H_0}(\text{rejeter } H_0)$. Ainsi, pour un α fixé, on rejettera H_0 si $p \leq \alpha$.

1.4.2 Validité d'un test et contrôle de l'erreur

Quand on réalise un test, on peut commettre deux types d'erreurs. On peut rejeter H_0 alors que l'hypothèse est vraie, on appelle cela une erreur de type I et on a alors un faux positif. On peut également faire une erreur de type II, c'est-à-dire ne pas rejeter H_0 alors que celle-ci est fausse et avoir un faux négatif. Ces erreurs et leur probabilité sont représentées dans la table 1.1.

Le contrôle de l'erreur de type I, aussi dit *validité* du test, est obtenu par construction et dépend du fait que les observations testées suivent bien les hypothèses intrinsèques au test (relatives à la distribution des observations par exemple). On peut vérifier qu'un test est valide au niveau α en simulant des observations sous l'hypothèse H_0 puis en vérifiant que $\mathbb{P}_{H_0}(\text{rejeter } H_0) \leq \alpha$.

Le contrôle de l'erreur de type II au niveau β assure un niveau de puissance $1 - \beta$.

La puissance d'un test est sa capacité à rejeter H_0 lorsque H_1 est vraie. Ainsi, on pourra évaluer la puissance d'un test en simulant des observations sous l'hypothèse alternative H_1 et évaluer la probabilité $1-\beta = \mathbb{P}_{H_1}(\text{rejeter } H_0)$.

Bien qu'elles ne soient pas symétriques, ces deux erreurs sont liées. En choisissant un α petit, on réduira le probabilité d'avoir un faux positif en rejetant moins « aisément » H_0. *A contrario*, on augmentera la probabilité d'avoir un faux négatif et donc β, entraînant ainsi une diminution de la puissance de détection du test ! Ainsi, si l'on a le choix entre deux tests valides pour nos données, on choisira celui avec la meilleure puissance.

Décision \ Vérité	H_0 est vraie	H_0 est fausse
H_0 est rejetée	Faux positif α	Vrai positif $1-\beta$
H_0 n'est pas rejetée	Vrai négatif $1-\alpha$	Faux négatif β

TABLE 1.1 – Représentation des quatre situations de test en fonction de H_0 (vraie ou fausse) et de la décision (rejet ou non) ainsi que de la probabilité associée.

Il est important de noter une différence concernant la loi des hypothèses H_0 et H_1. L'hypothèse H_0, qui correspond généralement à l'absence d'effet (et c'est le cas dans l'exemple spécifique de l'analyse différentielle), se caractérise plutôt bien. On peut ainsi obtenir la loi de la statistique de test sous H_0 et ce bien plus facilement que sous H_1. En effet, l'hypothèse alternative H_1 correspond alors à la présence d'un effet – possiblement arbitraire – et est généralement non spécifiée.

1.4.3 Tests multiples

Nous venons de décrire les deux erreurs que l'on rencontre dans la réalisation d'un test statistique et que l'on cherche à contrôler : l'erreur de type I et l'erreur de type II. Dans notre contexte, on réalise plusieurs tests statistiques à la fois, c'est une situation particulièrement courante en ce qui concerne les données en grande dimension telles que les données -omiques [Goeman and Solari, 2014] (génomique, protéomique, transcriptomique, etc.). Il est alors nécessaire de se placer dans un cadre de contrôle d'erreurs adapté à la réalisation d'un grand nombre de tests.

On peut montrer qu'en réalisant m tests à la fois, la probabilité de commettre au moins une erreur de type I dans le cas où les p-valeurs sont indépendantes est égale à $1-(1-\alpha)^{|H_0|}$ où $|H_0|$, qui est typiquement du même ordre de grandeur que le nombre total m de tests, est le nombre (inconnu) d'hypothèses nulles vraies. Cette probabilité tend donc rapidement vers 1 à mesure que le nombre de tests réalisés m augmente. Ainsi, le contrôle individuel de l'erreur de type I assuré par la validité du test ne suffit pas lorsque l'on réalise plusieurs tests à la fois.

La réalisation de m tests conduit à l'obtention de m p-valeurs $p_1, \cdots, p_m$. On prend ensuite une décision relativement au rejet de ces p-valeurs en fonction d'un certain seuil (généralement choisi égal à α) et on obtient alors un ensemble de p-valeurs rejetées noté

R. Il existe diverses stratégies pour assurer une garantie statistique globale dans le cadre des tests multiples. Ces approches permettent de corriger les p-valeurs (appelées alors p-valeurs ajustées) de manière à contrôler une erreur globale sur l'ensemble des tests. Ces stratégies diffèrent aussi bien en termes de type d'erreur contrôlée que de garantie fournie ou encore d'hypothèse faite sur la structure de dépendance entre les p-valeurs. Les deux erreurs les plus fréquemment utilisées dans le cadre des tests multiples sont le FWER (*Family-Wise Error Rate*) et le FDR (*False Discovery Rate*).

Le FWER correspond à $\mathbb{P}(|R \cap H_0| \geq 1)$, la probabilité que l'ensemble R contienne au moins un faux positif. Le contrôle de cette quantité peut être obtenu par plusieurs procédures dont la plus répandue est celle de Bonferroni [Bonferroni, 1936]. Elle consiste à définir des p-valeurs ajustées $\tilde{p}_j = \min(m \cdot p_j, 1)$ avec $j = 1, \cdots, m$ puis à prendre une décision concernant leur rejet en seuillant au risque choisi α. Cette approche a l'avantage d'être simple et valide quelle que soit la structure de dépendance entre les p-valeurs. C'est également le cas de la procédure de [Holm, 1979], qui est toujours au moins aussi puissante que la procédure de Bonferroni et doit donc lui être préférée. Plusieurs autres procédures comme celle de Hochberg [1988] ou Hommel [1988] peuvent permettre de gagner en puissance, au prix d'hypothèses plus restrictives sur les dépendances entre les p-valeurs. Cependant, la garantie requise par le FWER étant assez stricte, on pourra lui préférer un contrôle du FDR.

Le FDR correspond à $\mathbb{E}(\text{FDP})$ avec $\text{FDP} = |R \cap H_0|/|R|$. Le contrôle du FDR à un niveau α assure qu'en moyenne (relativement à l'ensemble des expériences que l'on pourrait réaliser), la proportion d'hypothèses rejetées à tort est bornée par α. Une des approches les plus largement utilisées pour le contrôle du FDR est la correction de Benjamini-Hochberg [Benjamini and Hochberg, 1995]. Après avoir ordonné les p-valeurs de manière croissante, $p_{(1)} \leq \cdots \leq p_{(m)}$, on calcule les p-valeurs ajustées comme $\tilde{p}_{(j)} = \min\left\{\min_{j \leq k}\left\{p_{(k)}\frac{m}{k}\right\}, 1\right\}$ [Yekutieli and Benjamini, 1999]. Cette correction est valide dès lors que les p-valeurs sont indépendantes ou bien présentent une dépendance positive dite PRDS (*Positive Regression Dependency on each one from a Subset*) [Benjamini and Yekutieli, 2001].

1.4.4 Application à l'analyse différentielle de données Hi-C

Les méthodes d'analyse différentielle de données Hi-C basées sur les pixels consistent à réaliser un test statistique pour chaque pixel (i, j) afin de déterminer s'il est différentiel. L'hypothèse H_0^{ij} sous-jacente est de la forme : « Le nombre d'interactions enregistrées entre les bins i et j n'est pas différent entre les deux conditions biologiques $\mathcal{C}_1$ et $\mathcal{C}_2$ ».

Les différentes approches existantes pour réaliser ce type de test sont décrites et comparées dans le chapitre 2. Nombre de ces approches consistent à supposer que les observations suivent une certaine distribution adaptée aux données de comptage comme la loi négative binomiale. On a alors $m_{ij}^{g,l} \sim \text{NB}(\lambda_{ij}^g, \phi_{ij})$ où le paramètre de dispersion ϕ_{ij} est propre au pixel, permettant ainsi de prendre en compte la variabilité intra-condition. Comme proposé dans Lun and Smyth [2015], on utilise un modèle linéaire généralisé pour obtenir : $\log(\lambda_{ij}^g) = \beta_{ij}^{(0)} + \beta_{ij}^{(1)}\mathbf{1}_{[g=2]} + \log(N_{ij}^{g,l})$ où $N_{ij}^{g,l}$ dépend de la taille de la librairie et de facteurs de normalisation (obtenus avec une normalisation MA par exemple). Pour chaque pixel (i, j), l'hypothèse H_0 s'écrit alors « $H_0^{ij} : \beta_{ij}^{(1)} = 0$ », ce qui permet de tester l'effet

de la condition sur le nombre d'interactions. Cela revient à faire une hypothèse d'égalité des moyennes entre les deux conditions, de la forme « $H_0^{ij} : \lambda_{ij}^1 = \lambda_{ij}^2$ » où λ_{ij}^g est la moyenne de la condition g dans la population.

Les approches utilisées ici dans le cadre de l'analyse différentielle de données Hi-C sont similaires à celles appliquées pour l'analyse d'expression différentielle de gènes [Chen et al., 2025, Love et al., 2014].

La réalisation de ces tests conduit à l'obtention d'un grand nombre de p-valeurs. En effet, si on considère un ensemble de matrices Hi-C correspondant à un chromosome découpé en d bins, il peut y avoir jusqu'à $d(d+1)/2$ p-valeurs issues des tests. Ainsi, avec $d = 2000$ bins, on réalisera deux millions de tests. Dans ce cas, le contrôle pour la multiplicité des tests est typiquement réalisé par l'approche Benjamini-Hochberg. Cette stratégie est valide en faisant l'hypothèse PRDS sur les p-valeurs, hypothèse communément admise pour des données de génomique [Goeman and Solari, 2014].

La correction pour la multiplicité est d'autant plus sévère que le nombre de pixels testés est élevé. Ainsi, afin d'atténuer cet effet et de gagner en puissance de détection, il est d'intérêt d'avoir un nombre plus faible de tests réalisés. Cela peut se faire – et c'est proposé par certaines méthodes de ce type – par le filtrage de pixels avec comme objectif d'enlever ceux avec des faibles comptages, peu probablement différentiels. Cependant, comme illustré dans Bourgon et al. [2010], ce filtrage doit être indépendant de la statistique de test, sous peine d'obtenir des p-valeurs « optimistes » et ainsi produire un excès de faux positifs.

Comme discuté dans la section 1.3, les approches existantes d'analyse différentielle de données Hi-C – basées sur les structures ou sur les pixels – ont de nombreuses limites. Dans le premier cas, elles imposent de fortes contraintes sur la structure des régions différentielles potentiellement identifiables et ne fournissent pas toujours de garanties statistiques satisfaisantes. Dans le second cas, l'interprétation des résultats est limitée à l'identification ponctuelle et locale de pixels différentiels. Afin d'aller au-delà des limites de ces deux types d'approche, on pourrait vouloir utiliser les résultats de l'analyse différentielle basée sur les pixels pour former des régions correspondant à des ensembles de p-valeurs ajustées significatives contiguës sur la carte Hi-C. Or, comme cela a été illustré dans Ebrahimpoor and Goeman [2021], affirmer qu'un ensemble de pixels correspond à une région différentielle car les pixels ont été sélectionnés à partir de p-valeurs ajustées significatives n'est pas correct. En effet, la garantie donnée par la correction de Benjamini-Hochberg porte sur l'ensemble des pixels en contrôlant (l'espérance de) la proportion d'hypothèses rejetées à tort. Ainsi, ce contrôle global n'est plus assuré sur un sous-ensemble de pixels.

1.5 Inférence *post hoc*

Comme explicité dans la section précédente, à l'issue de la procédure de Benjamini-Hochberg, on obtient un sous-ensemble de pixels R qui correspond aux hypothèses rejetées au seuil α à l'aide des p-valeurs ajustées. R est le plus grand sous-ensemble tel que $\mathrm{FDR}(R) \leq \alpha$. Avec ce type de procédure, on ne possède aucune garantie sur un quelconque sous-ensemble de p-valeurs différent de R. Dans le cadre de l'analyse différentielle de

données Hi-C, ce type de procédure apporte une interprétabilité limitée. En effet, on cherche à identifier des régions de la carte Hi-C qui représentent une réorganisation de la conformation tridimensionnelle entre deux conditions. De ce fait, on souhaiterait plutôt obtenir des garanties sur différentes régions de la matrice qui correspondent à plusieurs sous-ensembles de pixels et donc de p-valeurs.

Le cadre proposé par les méthodes d'inférence *post hoc* permet de répondre à cette problématique en donnant des garanties sur le nombre de faux positifs dans des sous-ensembles de p-valeurs arbitrairement sélectionnés. Plus précisément, l'objectif des méthodes *post hoc* est de fournir simultanément, pour plusieurs sous-ensembles de p-valeurs (qui peuvent être imbriqués ou chevauchant), une majoration du nombre de faux positifs dans chacun d'entre eux [Blanchard et al., 2020]. L'aspect particulièrement intéressant de ces stratégies est qu'elles sont applicables quel que soit le type de sélection utilisé sur les résultats et donc pour tout sous-ensemble de p-valeurs d'intérêt.

Formellement, on considère m p-valeurs $p_1, \cdots, p_m$ et un sous-ensemble quelconque $C_k \subset \{p_1, \cdots, p_m\}$. L'inférence *post hoc* permet de borner (ici, majorer) $|C_k \cap H_0|$, le nombre de faux positifs dans C_k. Cette borne, que l'on appelle borne *post hoc*, est une fonction V_α telle que

$$\mathbb{P}(\forall C_k, |C_k \cap H_0| \leq V_\alpha(C_k)) \geq 1 - \alpha. \tag{1.1}$$

Ainsi, on interprète l'équation (1.1) comme : « Avec probabilité au moins $1-\alpha$, quel que soit le choix de C_k, le nombre de faux positifs dans C_k est borné par $V_\alpha(C_k)$ ». De manière analogue, comme $\mathrm{FDP}(C_k) = |C_k \cap H_0|/|C_k|$, cette quantité est aussi bornée par $V_\alpha(C_k)/|C_k|$.

L'obtention de tels résultats repose sur l'existence d'une borne *post hoc* valide sur les données considérées, c'est-à-dire d'une fonction $V_\alpha(\cdot)$ qui vérifie l'équation (1.1).

Différentes bornes *post hoc* ont été développées dans de nombreux contextes tels que la réalisation de tests paramétriques sous diverses hypothèses de dépendance [Blanchard et al., 2020, Goeman and Solari, 2011], le test d'hypothèses structurées [Durand et al., 2020] (par exemple pour des hypothèses organisées de manière hiérarchique), dans le cadre de la randomisation des p-valeurs pour des tests à un ou deux échantillons [Blain et al., 2022], etc. Notons que ce type d'approche a déjà été appliqué à des données -omiques [Enjalbert-Courrech and Neuvial, 2022].

Parmi ces bornes, la borne de Simes, construite à partir de l'inégalité du même nom [Simes, 1986], s'écrit sous la forme :

$$V_\alpha^{Simes}(C_k) = \min_{1 \leq n \leq |C_k|} \left[\sum_{p_j \in C_k} \mathbf{1}_{\{p_j > \frac{\alpha n}{m}\}} + n - 1 \right]. \tag{1.2}$$

Cette borne est valide sous une hypothèse d'indépendance des p-valeurs ou encore de dépendance positive (PRDS). Ces hypothèses sont les mêmes que celles qui assurent la validité du contrôle du FDR par la procédure de Benjamini-Hochberg. Ainsi, dans un contexte dans lequel on utiliserait habituellement cette dernière procédure, on peut aisément gagner en interprétabilité en utilisant plutôt les outils et garanties fournis par

l'inférence *post hoc*. En particulier, tandis que le procédure de Benjamini-Hochberg fournit un unique sous-ensemble de p-valeurs R sur lequel le FDR = $\mathbb{E}$(FDP) est contrôlé, l'inférence *post hoc* donne, avec probabilité $1-\alpha$, un contrôle du FDP sur n'importe quels sous-ensembles $C_1, \cdots, C_K$ à travers la borne $V_\alpha(\cdot)$.

1.6 Contributions de la thèse

Durant ma thèse, je me suis intéressée à l'analyse différentielle de données Hi-C. Les deux types d'approches existants pour réaliser ce type d'analyse possèdent chacun leurs avantages et leurs défauts, ainsi que des problématiques qui leur sont communes. Pour les méthodes basées sur les pixels, les résultats peuvent manquer d'interprétabilité tandis que pour celles basées sur les structures, les méthodes ne fournissent pas toujours de garanties statistiques et se basent sur des définitions de structures qui ne sont pas consensuelles. La validation des résultats est une problématique commune à ces deux types d'analyse. En effet, il est compliqué de comparer et valider ces résultats car on ne possède pas de jeu de données de référence. De ce fait, ces derniers sont généralement validés sur un jeu de données annoté manuellement ou en recherchant la présence de marqueurs biologiques dans les régions différentielles identifiées. Par ailleurs, que ce soit pour les outils d'analyse différentielle basés sur les pixels ou sur les structures, il n'existe pas de benchmark qui permette de les comparer et de guider l'utilisatrice concernant le choix d'un outil. Les différentes limites identifiées ont conduit à la réalisation d'un travail en deux parties.

Dans une première partie, décrite dans le chapitre 2, j'ai participé à un travail qui a donné lieu à une publication visant à comparer des méthodes d'analyse différentielle basées sur les pixels. Cela a été motivé par le constat suivant : un grand nombre d'outils proposent de réaliser ce type d'analyse mais il n'existe pas de réelle comparaison permettant d'orienter l'utilisatrice vers une ou plusieurs méthodes à utiliser préférentiellement. Dans le papier, les méthodes sont décrites et comparées selon plusieurs critères relatifs au pré-traitement des données (normalisations et filtres), à la modélisation, aux sorties (p-valeurs et/ou p-valeurs ajustées) et aux caractéristiques techniques (format d'entrée des données, implémentation, etc.). Cette première description a permis de mettre en évidence la grande diversité des méthodes sur chacun de ces points. Les méthodes sont également objectivement comparées d'un point de vue statistique. La construction d'un cas H_0 a permis de créer deux conditions entre lesquelles aucune différence n'est attendue et ainsi évaluer la validité (contrôle de l'erreur de type I) des tests statistiques proposés. De même, grâce à la construction d'une région différentielle dans une région connue de la matrice Hi-C, il a été possible d'évaluer l'erreur de type II (puissance de détection) de chaque méthode. Enfin, les méthodes ont également été comparées et évaluées sur un jeu de données réel avec une validation de la pertinence de résultats par comparaison avec des données ChiP-Seq. Cette étude a permis d'établir une comparaison statistique rigoureuse des outils d'analyse différentielle basée sur les pixels.

Dans une seconde partie de la thèse, décrite dans le chapitre 3, je me suis intéressée au développement d'une méthode d'analyse de données Hi-C allant au-delà des limitations des approches actuellement proposées, c'est-à-dire la limitation de la détection de régions différentielles à des structures définies *a priori* ou ponctuelles (des pixels). La méthode

issue de ce travail – **hicream** – introduit un cadre global d'analyse différentielle de données Hi-C. Ce cadre s'articule en trois étapes. Une première étape consiste à réaliser une analyse différentielle basée sur les pixels, permettant d'obtenir une p-valeur par pixel. La seconde étape consiste en la définition de régions candidates. Ces régions peuvent être obtenues automatiquement à travers un clustering des données ou être définies par l'utilisatrice si celle-ci possède des connaissances *a priori* sur les données. Une dernière étape d'inférence *post hoc* permet d'évaluer les régions candidates et de donner une mesure interprétable de la proportion de pixels différentiels dans chaque région. Ce cadre est modulable et offre une grande liberté à l'utilisatrice à la fois dans le choix de la méthode fournissant les p-valeurs et la définition des régions candidates. La pertinence de l'approche proposée a été validée par application sur des données réelles et comparaison avec des données biologiques.

Chapitre 2

Analyse et évaluation des outils d'analyse différentielle de données Hi-C basée sur les pixels

Ce chapitre contient l'article Jorge et al. [2025] publié dans *Briefings in Bioinformatics*. Les annexes de ce chapitre se trouvent en fin de manuscrit, en annexe A. Un résumé de l'article en français est donné ci-dessous.

Motivation L'organisation tridimensionnelle du génome a un rôle important dans divers processus biologiques. La technique Hi-C est utilisée pour étudier les structures chromosomiques en quantifiant les proximités spatiales entre régions génomiques. Bien que de nombreux outils permettent de détecter des différences dans des données Hi-C issues de deux conditions biologiques, il n'existe pas de benchmark complet comparant et évaluant ces outils.

Résultats Cette étude propose une revue et un benchmark complets de 10 outils d'analyse différentielle de matrices Hi-C au niveau des comptages (pixels). Ce benchmark évalue les méthodes statistiques, la facilité d'utilisation et les performances (en termes de précision et de puissance) de ces outils, à partir de données Hi-C réelles et simulées. Les résultats révèlent une forte variabilité des performances entre les outils, soulignant l'impact considérable des prétraitements ainsi que la difficulté qu'ont tous les outils à contrôler efficacement le taux de faux positifs pour différentes résolutions et tailles de chromosomes.

Disponibilité Le code complet de l'étude est disponible sur `https://forgemia.inra.fr/scales/replication-chrocodiff` et utilise les données déposées sur `https://doi.org/10.57745/LR0W9R`.

Briefings in Bioinformatics, 2025, **26(2)**, bbaf074
https://doi.org/10.1093/bib/bbaf074
Review

A comprehensive review and benchmark of differential analysis tools for Hi-C data

Elise Jorge[1], Sylvain Foissac[1], Pierre Neuvial[2], Matthias Zytnicki[3], Nathalie Vialaneix[3,*]

[1]GenPhySE, Université de Toulouse, INRAE, ENVT, 31326 Castanet-Tolosan, France
[2]Institut de Mathématiques de Toulouse, UMR 5219, Université de Toulouse, CNRS UPS, 31062 Toulouse, France
[3]Université Fédérale de Toulouse, INRAE, MIAT, 31326 Castanet-Tolosan, France

*Corresponding author. Université Fédérale de Toulouse, INRAE, MIAT, 31326 Castanet-Tolosan, France. E-mail: nathalie.vialaneix@inrae.fr

Abstract

Motivation: The 3D organization of the genome plays a crucial role in various biological processes. Hi-C technology is widely used to investigate chromosome structures by quantifying 3D proximity between genomic regions. While numerous computational tools exist for detecting differences in Hi-C data between conditions, a comprehensive review and benchmark comparing their effectiveness is lacking. **Results:** This study offers a comprehensive review and benchmark of 10 generic tools for differential analysis of Hi-C matrices at the interaction count level. The benchmark assesses the statistical methods, usability, and performance (in terms of precision and power) of these tools, using both real and simulated Hi-C data. Results reveal a striking variability in performance among the tools, highlighting the substantial impact of preprocessing filters and the difficulty all tools encounter in effectively controlling the false discovery rate across varying resolutions and chromosome sizes. **Availability:** The complete benchmark is available at https://forgemia.inra.fr/scales/replication-chrocodiff using processed data deposited at https://doi.org/10.57745/LR0W9R. **Contact:** nathalie.vialaneix@inrae.fr

Keywords: Hi-C; differential analysis; statistical tests; benchmark

Introduction

Chromosomes are highly compacted within the cell nucleus, resulting in the spatial proximity of linearly distant genomic positions [1]. Hi-C [2] is a widely used technology to profile the 3D organization of the genome. It does so by estimating the spatial proximity between pairs of genomic positions through their frequency of interaction. The typical output of a Hi-C experiment, after preliminary data preprocessing, is usually summarized as a symmetric matrix of counts, where the entry (i, j) (or (j, i)) corresponds to the number of interactions registered during the Hi-C experiment between genomic regions ("bins") i and j. Hi-C has been widely used to uncover structural genomic elements at different hierarchical levels, such as A/B chromatin compartments, TADs, and loops [1–3]. Many computational tools exist to call these structures from Hi-C data, with variable reliability however [4–6].

Changes in 3D structures have been implicated in gene expression, cell division, cell differentiation, developmental disorders, and cancers [7–9]. This underscores the need for reliable methods and tools to compare Hi-C data across different conditions. One approach to comparing Hi-C data is to compute a similarity score for a pair of matrices, either at the level of the entire matrix (matrix-level) or for specific genomic regions (bin-level). Gunsalus et al., 2023 [10] reviewed several methods for the pairwise comparison of Hi-C matrices, classifying them into three categories: *basic methods*, which directly compute a similarity score (e.g. a correlation) between two matrices [11], *map-informed methods*, which first calculate a Hi-C-related metric along a 1D track for each matrix separately (e.g. directionality index) and then compare the resulting tracks [12], and *feature-informed methods*, which predict specific chromatin structures for each matrix (e.g. TAD boundaries or chromatin loops) before comparing the predictions [13]. While these methods offer various similarity or dissimilarity metrics, none provide statistical guarantees such as p-values. Moreover, they focus solely on pairwise matrix comparisons without incorporating biological replicates.

Another approach to comparing Hi-C data is differential analysis. Instead of quantifying the overall similarity between two Hi-C matrices (one for each condition), differential analysis aims at identifying local differences with statistical guarantees, often leveraging biological replicates for each condition. Following the previous classification, some of the methods can be considered as *map-informed*, as they use 1D metrics to detect differential structures such as TAD boundaries [14, 15] or chromatin compartments [16] at the bin level. Other methods fall under the *feature-informed* category, aiming to identify differential TADs, for instance [17]. However, most tools for differential analysis of Hi-C data do not fit into these categories, as they test for differences at the level of the bin pair, focusing on interaction counts between genomic regions [18–25]. To our knowledge, these methods have not been

Received: September 26, 2024. **Revised:** January 24, 2025. **Accepted:** February 10, 2025
© The Author(s) 2025. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf074/8051526 by guest on 07 May 2026

extensively reviewed or benchmarked. A recent book chapter [26] describes a few (four) such methods. While it provides technical instructions on their use and visualization of results, it does not evaluate the quality of the results.

To address this gap, we propose a comprehensive review of the following tools for the differential analysis of Hi-C data: **ACCOST** [18], **CHESS** [27], **diffHic** [19], **FIND** [20], **HiCcompare** [21], **HiCDCPlus** [22], **multiHiCcompare** [23], **Selfish** [24], and **sslHiC** [25]. We also considered a former version of **HOMER** (The version available and documented at http://homer.ucsd.edu/homer/interactions/.), which included a test to perform comparisons between two matrices, although 3D structure changes are not the primary focus of that tool. Our review provides a detailed technical description of each tool, focusing on implementation aspects, usability, and scalability. We explain the differences between the statistical methods employed by these tools and analyze their expected impacts on the results.

We also conducted two extensive benchmarks of the tools using real Hi-C data from the literature. The first benchmark used Hi-C data generated from a human tissue sample, with an artificially introduced ground truth to allow a quantitative evaluation of each tool's precision and power. The second benchmark involved Hi-C data from a CTCF depletion study during mouse cell cycle progression, evaluating the biological relevance of each tool's results by comparing them to findings from ChIP-seq experiments.

The article is organized as follows: The second section "Methods" reviews the statistical grounds of the different tools in a rigorous way. The third section "Implementation and usability" describes the technical aspects of the tools. The fourth section "Numerical experiments" introduces our benchmark protocol and the fifth section "Results" analyses the tools' performances.

Methods

Methodological overview of the tools

This article covers tools that all aim at answering the same question: given a set of n Hi-C matrices, $\mathcal{M}_1, \ldots, \mathcal{M}_n$, belonging to K different groups of biological interest (that we will call "conditions"), are we able to find bin pairs with significantly different interaction counts between conditions? While several descriptive metrics (such as correlation or other similarity measures) can be used for this purpose, we focus on approaches that provide statistical guarantees for identified bin pairs. Such approaches perform one statistical test for each bin pair. The result of each of these tests can be summarized by a p-value (or an adjusted p-value), which quantifies the statistical evidence of a significant difference.

The tools discussed in this article all have a common workflow (Fig. 1). In short, this workflow takes Hi-C matrices from different conditions and performs a statistical test, which results in a p-value (or an adjusted p-value) for each bin pair. **CHESS** is the only tool that slightly differs from this description because it provides p-values for fixed-sized windows of the Hi-C matrix (and not for every bin pair; see Section "Methodological background of the tools").

As shown in Fig. 1, the Hi-C differential analysis workflow can be decomposed into four main steps:

- *filtering*, which consists in removing bin pairs considered not relevant from the analysis in all Hi-C matrices;
- *normalization*, which consists in making bin pairs in a matrix or bin pairs between different matrices more comparable;
- *model and p-value computation*, which is the core of the statistical analysis and performs a test on all remaining bin pairs, using normalized interaction values;
- *multiple testing correction*, which aims at accounting for the fact that a large number of tests have been performed.

Most tools operate at the chromosome level, detecting intra-chromosomal (*cis*) differential interactions only. However, **diffHic**, **HOMER**, and **sslHiC** can also detect inter-chromosomal (*trans*) interactions.

The steps of the Hi-C differential analysis workflow and the various options used by different tools are described in detail in the sections below. Table 1 summarizes the main methodological characteristics of the tested tools in relation to the steps of this workflow.

Filtering

Several tools propose to remove some bin pairs before the analysis. The rationale behind this step is that low quality bin pairs or bin pairs with low interaction counts have little (if any) chance to be identified as differential but increase the number of hypothesis tests that are performed. Including such bin pairs can affect the test power, due to stronger multiple testing correction (see corresponding section below for further details). Discarding bin pairs before the test (and independently from its result) is a standard way to reduce this impact [28, 29].

The most common filters used in Hi-C differential analyses are:

- *low count filtering* (implemented in **diffHic**, **HiCcompare**, **HiCDCPlus**, **multiHiCcompare**, and **Selfish**). These filters simply remove, from the analysis, bin pairs that have interaction counts below a certain threshold. This threshold is either user-defined (i.e. all bin pairs for which the average or total interaction counts across analyzed matrices is below the threshold as in **diffHic**, **HiCcompare**, **multiHiCcompare**, and **Selfish**) or data-driven (the threshold is obtained as an estimation of a "background signal" from the data as in **diffHic** and **HiCDCPlus**);
- *bin quality filtering* (implemented in **CHESS**, **ACCOST**, and **HiCDCPlus**). These filters remove bin pairs including at least one bin with low mappability or, alternatively, bin pairs for which the interaction counts are below the expected (data-driven) value considering the bin mappability and GC content. These filters require that mappability or GC content is provided for each bin.

Additionally, **Selfish** allows to discard all bin pairs for which the genomic distance between the two bins is larger than a certain (user-defined) value, typically targeting bins with low interaction counts. Also, **HiCcompare** has an option to let users specify a set of bins that should not be considered in the analysis. Supplementary Table S1 summarizes the type of filters available in each tool.

Normalization

Hi-C matrix normalization is an important step of the workflow. It aims at removing technical or biological biases that can impede a fair comparison between bins or between matrices [30]. The main known biases that can affect Hi-C data analysis are scale differences between matrices (e.g. due to differences in sequencing depths), scale differences between bins in a given matrix (e.g. due to differences in mappability), or effects of the genomic distance in interaction counts within a given matrix. In the current section,

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf074/8051526 by guest on 07 May 2026

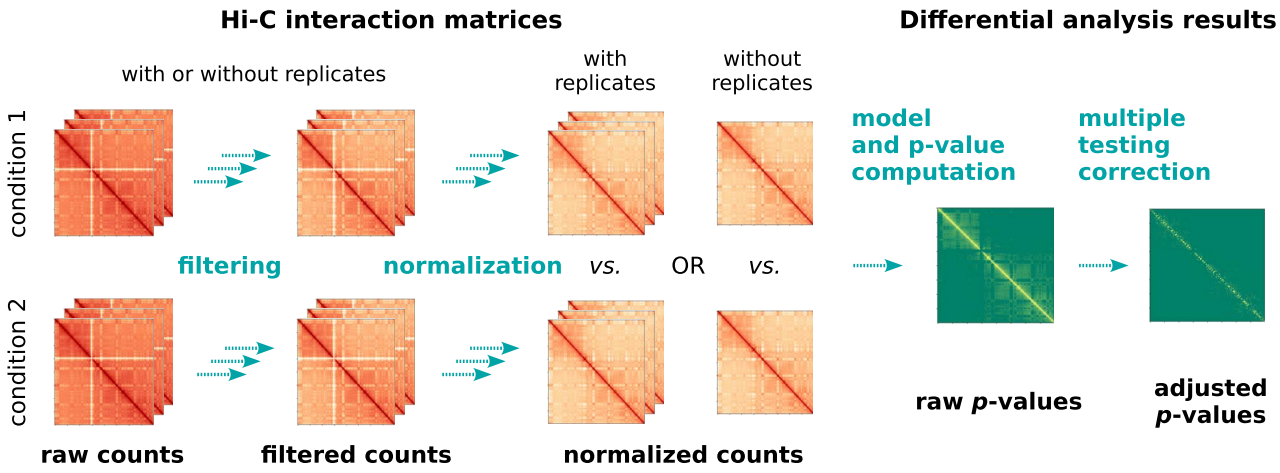


Figure 1. A schematic representation of the typical workflow for differential analysis of Hi-C matrices at the bin-pair (pixel) level. Input matrices from two conditions, with or without replicates, are first filtered and normalized (first two steps). Statistical tests are then conducted to generate raw *p*-values, which are subsequently adjusted for multiple testing correction (last two steps). Additionally, a (log) fold change matrix can be generated, representing the ratio of average interaction values between conditions for each bin pair (not shown).

we discuss these biases, their impact on the analysis, and how the different tools address them.

Differences in total interaction counts between matrices. Total interaction counts across all bin pairs can differ between matrices due to experimental factors, such as variations in sequencing depth or library complexity. To prevent false positive predictions that would incorrectly label bin pairs as "differential", these technical artifacts must be accounted for, as commonly done in RNA-seq [31] or ChIP-seq data [32, 33] differential analyses.

The most straightforward method to correct this bias is the total sum scaling (TSS) that simply aligns the total counts of all matrices in the dataset (implemented in **sslHiC** and advised, but not implemented, in **HiCcompare**). However, this approach has been shown to be generally inefficient for sequencing data, as it is strongly influenced by large outlier counts [31].

Hence, the article of Lun & Smyth, 2016 [33] emphasizes the need for an adequate between-matrix normalization and proposes the MA correction (correction of the trend in an MA plot, where the difference "M" between two or more matrices is displayed as a function of their average count "A"). This correction, performed by cyclic locally estimated scatterplot smoothing (LOESS), has been shown to be efficient for ChIP-seq data and robust to a large proportion of low counts. It is implemented in **diffHic** (and advised in **FIND** but not implemented).

A more sophisticated alternative is used in **HiCcompare** and **multiHiCcompare**. The MA correction is replaced by an MD correction (where D stands for the genomic distance between bin pairs, instead of its average count). However, since A and D are strongly related in Hi-C matrices (the larger the distance between the two bins of a pair, the lower the count for this bin pair), both methods are expected to result in similar corrections.

Finally, although this does not strictly aim at correcting differences in sequencing depths, **sslHiC** also implements a min/max normalization applied to $\log_{10}$-transformed matrices so as to make all counts in a given matrix lie between 0 and 1 (and thus be more comparable across matrices).

Differences in total counts between bins within a given matrix. The total number of interactions assigned to a specific bin or over a given genomic region depends on local properties of the genomic sequence, such as GC content, mappability, or restriction site density [34]. Since the purpose of differential analysis is to compare bin pairs between matrices and not bin pairs within the same matrices, correcting for these biases is not strictly necessary from a statistical perspective. However, several tools nevertheless recommend or implement methods for correcting these biases.

Among the most popular methods for within-matrix normalization, non-parametric methods do not explicitly use GC content or mappability values to remove biases between bin counts. On the contrary, similarly to TSS normalization, they align the observed total count across all bins of a given matrix. These include iterative correction and eigenvector decomposition (ICE) [35], implemented in **HOMER**, or Knight–Ruiz (KR) matrix balancing [36], implemented in **diffHic**. Other tools (**FIND** and **Selfish**) benefit from the juicer data format [37] and embed values that allow for KR correction. Finally, **CHESS** and **HiCDCPlus** recommend the ICE correction but do not implement it.

An alternative to choosing a specific method to correct between-bin biases is to let users provide bin-specific correction values. This is the course of action taken by **ACCOST** (that recommends using ICE but allows for any other type of bin correction values to be used).

Finally, note that if all bin sums in a given matrix are aligned to the same total count B (e.g. $B = 1$ for KR correction), then it is sufficient to use the same B for all matrices to align, at the same time, overall total interaction counts between matrices.

Genomic distance related differences between interaction counts within a given matrix. Hi-C matrices are strongly structured with respect to the genomic distance between bin pairs. Likewise biases between bin total counts within a given matrix, this bias does not necessarily require correction but is nevertheless accounted for in several tools.

One of the most popular approach to correct for this bias is to compute an "observed over expected" matrix. The interaction count of a bin pair is divided by the average interaction counts of all bin pairs with the same genomic distance. This approach is implemented in **CHESS**. Similarly, the interaction count of a bin pair at the same genomic distance can be centered and reduced to unit variance (as implemented in **Selfish**) or scaling factors for bin pairs at the same genomic distance (e.g. median) can be computed and used for normalization (as implemented in **ACCOST** and **HiCDCPlus**).

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf074/8051526 by guest on 07 May 2026

Table 1. Methodological description of the tools (summarized; for further details see the text)

	Preprocessing		Model					p-values	
Tool	**Filtering**	**Normalization**	**Data distribution assumption**	**Leverages replicates**	**2D-aware approach**	**Allows covariates or $K > 2$ conditions**	**Type of model/test**	**Provides raw p-values**	**p-value correction**
ACCOST	Bin quality	(ICE)	NB	✓	×	×	exact test	✓	(BH)
CHESS	Bin quality	(ICE)	None	×	✓	×	SSIM	✓	×
diffHic	Low counts	MA, KR	NB	✓	×	✓	GLM	✓	BH
FIND	×	(MA), KR	SPP	✓	✓	×	exact test	×	BH by distance
HiCcompare	User-defined	TSS, MD	None	×	×	×	Z-score	✓	BH by distance
HiCDCPlus	Low counts, bin quality	(ICE)	NB	✓	×	×	exact test	✓	BH
HOMER	×	ICE	NB	✓	×	×	GLM	✓	BH
multiHiCcompare	Low counts	MD	NB	✓	×	✓	GLM	✓	BH by distance
Selfish	Low counts, distance	KR	None	×	✓	×	Z-score	×	BH
sslHiC	×	TSS, min/max	None	×	✓	×	GNN	×	BH

ICE: iterative correction and eigenvector decomposition; KR: Knight–Ruiz; NB: negative binomial; SPP: spatial Poisson process; SSIM: structured similarity index; GLM: generalized linear model; GNN: graph neural network; BH: Benjamini–Hochberg. Items between parentheses correspond to steps advised by the method authors but not implemented in their tool

Other normalizations.
Other methods designed to correct various other biases are also implemented: **diffHic** proposes a method based on DNA copy number variation (CNV) estimation to correct for this bias. However, as discussed by Servant et al., 2018 [38], CNV could be of interest in cancer studies and it is therefore not necessarily sound to always use this correction.

Finally, note that all tools that use the genomic distance between bins for the normalization are restricted to detect intra-chromosomal (*cis*) differential interactions only and cannot consider inter-chromosomal (*trans*) interactions.

Supplementary Table S2 summarizes the different normalization options offered by the tools.

Methodological background of the tools

This section describes the methodological premises and solutions of the different tools. The tools can be classified according to the following two questions:

- Can the tool *use biological replicates* to perform the test (i.e. handle more than one matrix in each condition)? When biological replicates are available, it is still possible to use a tool designed to only compare one matrix in each condition by merging (computing the sum of) the replicates of each condition. However, it is strongly advised, from a statistical perspective, that these replicates are used in order to relate the inter-condition variability to the intra-condition variability.
- Does the tool consider interaction counts as independent from each other, or does it try to take advantage of the fact that two bin pairs, (i, j) and (i', j') in the matrix, tend to have more similar interaction counts when they are close to each other (e.g. $|i - i'| + |j - j'|$ is "small")? We will use the term "2D-agnostic" for the tools that consider bin pairs independent and the term "2D-aware" for the tools that account for this property.

These two typology levels for the tools are provided in the two columns of Table 1 named "use of replicates" and "2D-aware". We now give a brief overview of each tool based on the answer to these two questions.

Comparison of two matrices

The tools designed to perform differential analysis between two matrices are: **HiCcompare** (2D-agnostic), **CHESS**, **Selfish**, and **sslHiC** (2D-aware).

2D-agnostic method. A 2D-agnostic method means that a measure of the difference between the two matrices is obtained at bin pair level and transformed into a Z score, from which a p-value is derived using the Gaussian distribution. More precisely, **HiCcompare** uses the M-value (log-fold change between the two matrices) of the interaction count to obtain a Z score.

2D-aware methods. Existing 2D-aware methods that perform tests between two matrices are based on different premises: **CHESS** first partitions the Hi-C matrix into fixed-size square submatrices and computes a structural similarity index (SSIM). This index is commonly used in imaging analysis to quantify the similarity between two matrices. It depends on the average signal in each submatrix, the variance of the signal in a given submatrix, and the signal correlation between the two submatrices. A p-value is then derived for each square from this index, quantifying the exceptionality of the observed index with respect to a background model.

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf074/8051526 by guest on 07 May 2026

Selfish performs a sort of "local smoothing" of the matrices: For each bin pair, it applies Gaussian filters centered at the bin pair, with increasing radius. The idea is to take advantage of the spatial self-similarity in contact maps to improve statistical evidence. Differences of the Gaussian filter evolutions between the two matrices are then assumed to be Gaussian, from which a p-value is derived for each radius. The final p-value is defined as the minimum radius-specific p-value across radii. However, since no multiple testing correction is applied at this stage, the resulting p-values are invalid.

Finally, **sslHiC** is based on a graph neural network (GNN) [39, 40]. The idea is to represent a Hi-C matrix as a graph in which bins are nodes and positive interaction counts are edges (weighted by the interaction count). Bin pairs of the form $(i, i+1)$ (linking two bins that are neighbors on the chromatin) are also linked with an edge to encode the genome structure in the Hi-C graph. The authors propose a new architecture of GNN, which they call "edge-enhanced GNN" (EEGNN) that aims at better exploiting the information carried by edges in the message passing process of the GNN. Using this architecture, all the bin pairs (i, j) in the matrix are represented by their embeddings $h^k_{(i,j)}$, d-dimensional vectors organized in different layers, k. The method is fully aware of the whole matrix since the embedding $h^k_{(i,j)}$ at layer k is passed to the other bin pairs that share a common node to compute their embeddings at layer $k+1$. The method finally derives a p-value for (i, j) by assuming Gaussian distribution of the Euclidean distance between embeddings $h^K_{(i,j)}$ of the two matrices in the last layer K.

Comparison of multiple matrices for each condition.

Tools that leverage replicates to perform differential analysis are **ACCOST**, **diffHic**, **HiCDCPlus**, **HOMER**, **multiHiCcompare** (2D-agnostic), and **FIND** (2D-aware). As explained above, an important advantage of these tools from a statistical perspective is that they can account for the variability across replicates within each condition (e.g. by computing variances, which cannot be done when a single replicate is available).

2D-agnostic methods. 2D-agnostic tools that account for replicates (**ACCOST**, **diffHic**, **HiCDCPlus**, **HOMER**, and **multiHiCcompare**) all assume that the interaction counts follow a negative binomial (NB) distribution. This is a standard hypothesis already used in other differential analysis methods for sequencing data, notably for RNA-seq. More precisely, **diffHic** and **multiHiCcompare** integrate **edgeR** [41] functions that fit a NB generalized linear model (GLM) (and thus directly benefit from the flexibility of this framework, able to account for complex experimental designs). The main difference with the standard RNA-seq pipelines is the addition of an offset derived from the MA (**diffHic**) or MD (**multiHiCcompare**) normalization in the NB GLM. Similarly, **HiCDCPlus** and **HOMER** use **DESeq2** [42] and differ from **DESeq2** by the preprocessing performed on Hi-C matrices, especially the filtering step described in the "Filtering" section. Alternatively, **HOMER** can also use an **edgeR** model. Even if it does not directly depend on **DESeq2**, **ACCOST** also derives its method from **DESeq2**'s NB model, plugging the bin-specific correction values described in "Normalization" into the NB GLM method of **DESeq2**.

2D-aware method. The only 2D-aware tool able to account for replicates is **FIND**. In **FIND**, a bin pair is described by its position (i, j) in the matrix 2D structure and its interaction counts across matrices. The resulting triplet is distributed as a spatial Poisson process (a count process that has a spatial structure) with condition-specific intensity parameter λ_1 and λ_2. A first-level p-value for the test of the null hypothesis $\lambda_1 = \lambda_2$ is then obtained for each bin pair. The final p-value at each bin pair is obtained by aggregating the first-level p-values in the local neighborhood around the bin pair, using the r-ordered p-value (rOP) method [43]. However, the resulting p-value may not be valid since the rOP method assumes independence between the hypotheses to be aggregated.

Multiple testing correction

All the evaluated tools perform one statistical test for each bin pair (i, j), with **FIND** and **Selfish** deriving this p-value by aggregating results from other previous tests. Therefore, a multiple testing correction is necessary to control false positives–bin pairs identified as differential by chance rather than due to a true difference in interaction levels between the two conditions. Multiple tests in genomic studies are generally handled by controlling the false discovery rate (FDR). The FDR corresponds to the expected proportion of false positives among the bin pairs called significant by a given method. The state-of-the art method for FDR control is the Benjamini–Hochberg (BH) method [44].

However, multiple testing correction is handled in different ways across tools. **diffHic**, **HiCDCPlus**, **HOMER,** and **sslHiC** implement FDR control using the BH method. While **ACCOST** does not directly provide multiple testing correction, its authors also used the BH method in [18]. A different strategy is implemented in **HiCcompare**, **multiHiCcompare**, and **FIND**. These methods perform multiple testing correction on a per-distance basis, also using the BH method. This implies that the FDR of their results is (theoretically) not globally controlled at the chromosome level, which means that more false positives can be expected for these tools. Notably, as the typical use case of the tools considers individual chromosomes, looking for differences in *cis*-interactions, multiple testing correction is not performed by these methods at the genome-level.

Handling more complex experimental designs

Finally, **diffHic** and **multiHiCcompare** are designed to perform a test between more than $K = 2$ conditions or are able to include external covariates in the model. The latter is useful when an experimental factor is not of primary interest for the differential analysis but might influence the results (e.g. a noise effect, like the sex or the tissue, could mask the differences due to the factor of interest, like the treatment). In this case, it is common practice to account for this covariate as a "blocking factor," correcting the effect without testing for it. However, due to the high cost of Hi-C data generation, such complex designs (involving more than two conditions and/or covariates) remain rare. As a result, while these features may be valuable for future experimental designs, they are not the primary focus at present.

Implementation and usability of the tools

Table 2 summarizes technical information for each tool, including the programming language, whether the tool is packaged and easy to install, which input formats are handled, whether a documentation is provided, and when it has been last updated.

Inputs and input formats

Almost all the tested tools assume that the raw sequencing reads have preliminary been processed with a Hi-C data analysis pipeline and consequently converted into interaction matrices. Notable exceptions are **diffHic**, which can also handle BAM or FASTQ files, and **HOMER**, which requires BAM or FASTQ files.

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf074/8051526 by guest on 07 May 2026

Table 2. Technical description of the tested tools

Tool	Language	Installation	Input format	Documentation	Last modification	Code repository
ACCOST [18]	Python2 & R	Python scripts	own tsv	Basic local HTML manual	2022-07	ACCOST[1]
CHESS [27]	Python3	pip	`.hic`/`.cool`/`.fanc`	Complete "readthedocs" documentation	2023-07	Github
diffHic [19] (v1.26.0)	R	Bioconductor	`GInteractions` R object	Manual + Bioconductor vignette	2024-03	Github
FIND [20] (v1.0.0)	R	source R package	HiC-PRO	Basic online manual + local vignette	2022-11	Bitbucket
HiCcompare [21] (v1.16.0)	R	Bioconductor	HiC-PRO/own tsv	Manual + Bioconductor vignette	2023-06	Github
HiCDCPlus [22] (v1.2.1)	R	Bioconductor	`GInteractions` R object	Manual + Bioconductor vignette	2022-06	Github
HOMER [51]	Perl & C++	Perl & C++ script	FASTQ reads	Complete website	(not mentioned)	none
multiHiCcompare [23] (v1.12.0)	R	Bioconductor	HiC-PRO/own tsv	Manual + Bioconductor vignette	2022-04	Github
Selfish [24]	Python3	pip/container	`.hic`/`.cool`/HiC-PRO/own tsv	Basic manual	2021-05	Github
sslHiC [25]	Python3	Python scripts + conda env	`.cool`/`.mtx`/`.npy`/`.npz`	Basic online manual	2023-04	Github

Dates of last modification were collected in November 2023. Version of the tool used in the benchmark (when provided by the tool) is indicated in the first column. [1] We found three repositories for **ACCOST** code, one on Bitbucket and two on Github (one of them has not been updated for 8 years). We contacted the authors who indicated us the repository to consider

During the construction of the interaction matrix, paired-end reads are usually mapped to a genomic reference sequence. Chromosomes are then discretized into fixed-size bins, and interaction matrices are obtained by counting for each bin pair the number of read pairs that link the corresponding bins. In short, interaction matrices are essentially symmetric square matrices with non-negative entries and many zeros.

Several file formats have been proposed to store these matrices, with different degrees of adoption. Although none of them has become the universal standard yet, a few are used by several tools. Such common formats include binary (and possibly compressed) formats, like the `.hic` [37], `.cool`, `.mcool` [45] and `.fanc` [46] formats, and text-based formats like the HiC-PRO [47] or BEDPE [48] formats. A majority of the tested tools (namely **CHESS**, **FIND**, **HiCcompare**, **multiHiCcompare**, **Selfish**, and **sslHiC**) use these standard formats (Table 2). Note that **sslHiC** can take as input a `.cool` file or a contact matrix file similar to the one generated by **HiC-PRO**. In the latter case, unlike the other tools, it does not require an index file but only the matrix resolution (only certain resolutions are allowed; see Methods). The matrix file is then given as an `.mtx` file in the Matrix Market format, or can directly be passed as a binary Python/**numpy** file (`.npy` or `.npz`).

The other tools use more specific formats. For instance, **ACCOST** requires a tab-delimited format file, with columns `<chr1> <mid1> <chr2> <mid2> <#reads>`, where `mid`i is the position of the middle of bin i ($i = 1, 2$), and `#reads` is the raw interaction count. **HiCDCPlus** and **diffHic** use the `GInteractions` Bioconductor class [49] as input format. **HOMER** is the only tool in the list that exclusively accepts raw reads as input, rather than interaction matrices. As a result, users must map the data with **HOMER**, making it incompatible with pre-existing matrices for differential analysis.

Furthermore, some tools require additional data with the interaction counts. **ACCOST** requires a bin-specific normalization score for each bin, which can be obtained with the ICE method [35], as implemented, e.g. in Bioconductor/**HiC-PRO** package [47] or in **Cooler** [45]. Note that **ACCOST** can accommodate any possible bin-specific bias as input, allowing to use parametric methods based on GC content, mappability, or restriction site density, as long as they provide a score for each bin. Additionally, **ACCOST** requires a mappability score for each bin, but this information is only used to discard some bins from the analysis.

Likewise, **HiCDCPlus** requires GC content information, but the tool can compute it internally as long as the corresponding genome is available from Bioconductor [50]. Optionally, mappability information can also be provided. In contrast to other tools, **CHESS** performs a test and derives a p-value only if a set of background regions, where no difference is expected between the two matrices, is provided. Otherwise, **CHESS** only computes similarity scores between the matrices and does not return a p-value.

Finally, some of the tools contain format converters, like **HiCcompare** and **multiHiCcompare** that provide functions to convert `.hic` and `.cool` files to their own internal format.

Of note, **sslHiC** is the only tool that restricts the bin size. Namely, it can only analyze Hi-C matrices at resolutions 10, 50, or 500 kb, because the authors trained their deep-learning models for these resolutions only.

Programming languages and packaging

Most of the tools reviewed are implemented in Python and/or R (Table 2), with the exception of **HOMER**. From a user perspective, availability through a package management system (like

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf074/8051526 by guest on 07 May 2026

pip, **conda**, or the CRAN repository) is highly valuable because dependencies are usually handled during the installation process, making it much easier to install compared to non-packaged tools. Bioconductor packaging [50] offers additional stability for several reasons: the code is extensively reviewed before acceptance, every release is tested on the three main operating systems, and extensive documentation is required (including a use case vignette). Python packages often rely on an external documentation website, which can be extensive and detailed (such as the ones hosted on the "Read the Docs" documentation service https://about.readthedocs.com/).

From this point of view, the R/Bioconductor packages **diffHic**, **HiCcompare**, **HiCDCPlus**, and **multiHiCcompare** are easy to install, thanks to the Bioconductor common installation process. **FIND** is also easy to install, even if not included in an official package repository.

Similarly, for Python tools, **CHESS** and **Selfish** are easy to install, thanks to **pip**. In addition, **Selfish** also proposes an installation process via a **Docker** or a **Singularity**/**Apptainer** container, providing further reproducibility and robustness. In contrast, **ACCOST** does not offer a **pip** installation and is just provided as Python scripts. For **ACCOST**, the authors simply mention its dependency with Python 2.7 and R, as well as with **numpy**, **scipy**, and some **scikit-learn** libraries. In contrast, **sslHiC** is easier to use thanks to a provided conda environment setting file. **HOMER** includes a script, which installs and configures the tool. A description of the documentation of the tools, together with their comprehensivenesses and readabilites, and a description of issues encountered during installation of the tools are provided as Supplementary Sections 3.1 and 3.2, respectively.

Illustrative datasets

In addition, having some data included in the tool for illustration is usually appreciated by users. From this perspective, **HOMER** and **Selfish** do not include any dataset. **diffHic** includes a small BAM file used in its manual, while its vignette features three external datasets also mentioned in their article. **FIND**, **HiCcompare**, **HiCDCPlus**, and **sslHiC** include (part of) the processed data from [3] (GEO: GSE12878), which they used in their documentation and also (except for **FIND**) in the results of the article. **ACCOST** also includes part of the same dataset but does not use it in the HTML manual for illustration (this dataset is discussed in their article, however). **sslHiC** also includes datasets simulated from chromosome 21 of a GM12878 cell line dataset (the original dataset is only used to illustrate another feature of the tool on a replication measure). The simulated dataset consists in three couples of matrices including a certain percentage of simulated differential interactions with varying fold changes (2, 4, and 6). **CHESS** includes the processed data from [52] (ArrayExpress: E-MTAB-5875) and **multiHiCcompare** part of the data from [53] (GEO: GSE104888). Both use these datasets in their documentation and article.

Note that all datasets are not provided under the same format. **ACCOST** provides compressed `.tsv` files that correspond to their input format, **CHESS** and **HiCDCPlus** provide `.hic` files, **FIND** and **HiCcompare** embed data in their tool (they can then be loaded using the function `data`, directly properly formatted for usage in their functions or as `GInteractions` objects [49]).

Numerical experiments

In this section, we present the extensive numerical experiments that we performed to assess the statistical performance of the tools. In particular, we describe the datasets, the tools, and how we designed the tests to evaluate the Type-I error control, the power, and the biological relevance of the results.

Tested tools

Among the tools described in "Methods," we excluded three tools from the simulation study:

1) **CHESS** because it is made to provide p-values for fixed-sized windows of the Hi-C matrix that "cannot be smaller than 20× the bin size of the data" (User documentation even recommends to use regions spanning at least 100× the bin size of the data.), which is hardly comparable with the other tools (that obtain results at a bin pair resolution);
2) **ACCOST** because it is not actively maintained anymore (it resulted in errors with Python. We contacted the authors about this problem which they intend to solve);
3) **HOMER**, which led to an error with our data. We contacted the authors about this problem without success.

Supplementary Table S3 provides the link to the source code and the version or date at which it was accessed for installation.

All tools were launched successfully for all experiments except for:

- **multiHiCcompare** that filtered out all bin pairs in chr 21 experiments with the semi-simulated dataset. All bin pairs were also filtered for chr 13, 14, and 15 of the CTCF depletion dataset. The tool was successful but no results were produced;
- **sslHiC** that we could run only on 500 kb resolution matrices. The tool was successful for this setting but not designed for the other settings.

Tool parameters

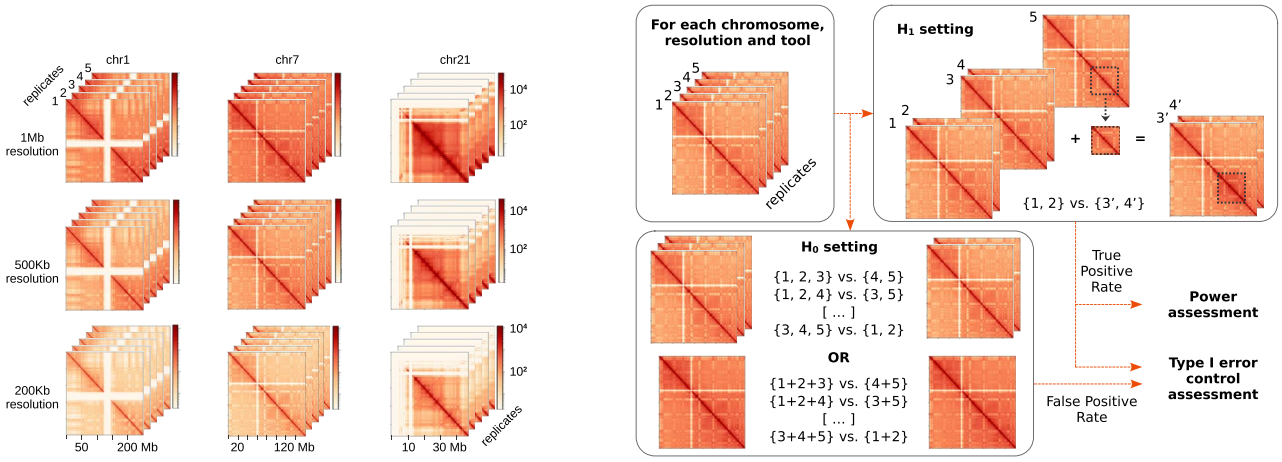
These tools were tested with their default parameters whenever possible. The exceptions to this choice are listed in Supplementary Table S4 and correspond to parameters with no default but required by the tool to work, as for **FIND**.

In addition, by default **FIND** filters out results for which the adjusted p-value was above a certain threshold. We turned this filter using `qvalue = 1` to retrieve all results (Using this setting results in **FIND** returning adjusted p-values equal to one as zeros, which is not desirable. We manually corrected this setting in our code.). We also used the option to split the computation into several chunks (otherwise, using the tool resulted in memory overload).

Similarly, **diffHic** provides several functions that can perform different types of filtering before the differential analysis. In our experiments, we did not filter out bin pairs with a low **log**CPM but we used their `filterTrended` filter.

For a given experiment and tool, p-values were adjusted independently for each chromosome. The BH procedure [44] was used to adjust p-values when the tool did not provide adjusted p-values. For tools that perform a per-distance-basis FDR correction (**HiCcompare** and **multiHiCcompare**), we kept their adjusted p-values and also computed adjusted p-values at the chromosome level ("standard" BH procedure). In the Results section, these two types of results are identified by **HiCcompare** (original adjusted p-value of the tool) and **HiCcompare-realFDR** (re-computed adjusted p-value). We were unable to perform this correction for **Selfish**, **FIND**, and **sslHiC**, which, unfortunately, do not provide raw p-values.

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf074/8051526 by guest on 07 May 2026



(a) Initial data (b) Simulation process

Figure 2. **Design of the simulations.** (a) The available matrices used for the numerical experiments consist in five technical replicates from three chromosomes (1, 7, and 21) generated at three different resolutions (200 kb, 500 kb, 1 Mb). Numbers at the bottom row correspond to genomic positions (in Mb), indicating the size of these matrices. (b) Illustration of the simulation process for Type-I error assessment (H_1 setting, top right) and assessment of false positive rate (FPR) and true positive rate (TPR) (H_0 setting, bottom left). Type-I error control was assessed by splitting technical replicates randomly into two groups, while FPR and TPR were assessed by generating artificial true positive examples where read counts are increased in a target zone by adding resampled Hi-C data from the remaining technical replicate.

Semi-simulated data study

We first used a use case where the ground truth of difference locations is precisely controlled. One possibility to do this would have been to rely on a simulation study, generating data from a specific probability distribution. A natural choice for this distribution would be the NB model, since a number of differential analysis tools rely on this distribution (**diffHic**, **HiCDCPlus**, **multiHiCcompare**, and **ACCOST**). However, the evaluation process would have then been biased in favor of these tools. More generally, any choice of a particular distribution induces biases since the true data generating distribution is unknown.

We therefore used semi-simulated data coming from real Hi-C data. This type of approach has previously been applied multiple times to benchmark tools for, e.g. RNA-seq data [54–56]. More specifically, we used an ENCODE dataset [57] from a Hi-C experiment performed on a human colon sample (experiment accession: ENCSR295BDK), that includes five technical replicates (sequencing runs). To obtain Hi-C matrices, raw sequencing reads of each technical replicate were processed using the `nf-core/hic` pipeline [58] v1.2.2 on the assembly version GRCh38 of the human genome (see Supplementary Section 4.1 for further details). Hi-C matrices at three different resolutions and for three different chromosomes were finally used, as shown in Fig. 2a. Processed data are available at https://doi.org/10.57745/LR0W9R.

To assess the Type-I error control, we ran each tool on technical replicates randomly split into two groups, where no signal is expected. We also assessed the statistical power of the tests by creating a controlled difference in a given part of some matrices. Figure 2 illustrates (a) the data matrices used and (b) the test protocol.

Type-I error control (H_0 setting)

The quality of statistical tests is usually assessed via their mathematical validity (proper control of the Type-I error, or false discoveries) and by their performance (statistical power or ability to detect true positives). In this first simulation setting, we generated data under the null hypothesis (H_0) in which no signal is expected, as described in Fig. 2b.

For each chromosome, we assigned the five technical replicates to two conditions (three replicates in a condition and the other two in the other condition) and processed them with the six tools to extract p-values for differential interactions between the two conditions. The $C_5^3 = 10$ possible assignments of the replicates into two groups were obtained and considered as independent experiments (i.e. p-values were adjusted independently in each assignment and chromosome).

For tools designed to compare only two matrices (one for each condition), i.e. **HiCcompare** and **Selfish**, we merged the replicates of the same condition into a single matrix before processing the two resulting matrices with the tool (Fig. 2b). Also, **Selfish** results were not symmetric (the p-value assigned to the bin pair (i, j) was not always equal to the p-value assigned to the pair (j, i) whereas the Hi-C matrix is symmetric by design and logFC were found identical between the two pairs). For instance, for simulation 6, chromosome 21, and resolution 1 Mb, **Selfish** returned a p-value of 0.9 for the pair $(2810, 2805)$ and a p-value of $7.4e^{-4}$ for the pair $(2805, 2810)$, as documented in our code repository). To address this, we arbitrarily kept one of the two p-values returned by the tool (the one corresponding to $i < j$).

The total number of performed tests, the percentage of significant results (based on p-values and adjusted p-values) at different risk levels as well as the empirical cumulative density function (ECDF) were obtained for each tool, chromosome, and resolution. Note that not all tools provided raw p-values. **FIND**, **Selfish**, and **sslHiC** only returned p-values adjusted for FDR control. For these tools, one can only verify that the average number of tests declared significant (at any target FDR level) is zero for a H_0 setting.

Simulations with ground truth signal (H_1 setting)

The same dataset was used to generate pseudo-simulated experiments corresponding to the existence of a region with a positive signal, as described in Fig. 2b. More specifically, the five technical replicates of each chromosome were used in the following way:

- two replicates were used as the Hi-C matrices of the first condition;

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf074/8051526 by guest on 07 May 2026

Table 3. Total number of bin pairs (third column) and number of bin pairs in the target zone for the evaluation of true positive detection rates (fourth column; H_1 setting only), for each chromosome and resolution

Chr.	Resolution	Total	In target
1	1 Mb	26 741	1275
7	1 Mb	12 765	861
21	1 Mb	861	45
1	500 kb	105 231	5151
7	500 kb	49 967	1888
21	500 kb	3218	136
1	200 kb	637 599	31 375
7	200 kb	306 209	10 936
21	200 kb	17 558	730

Note that, due to filters in the tools, not all these bin pairs were actually tested for each method.

- two other replicates were modified to be used as the Hi-C matrices of the second condition. We first selected a region, called "target zone", and increased the counts of the matrices of the second condition in this region by adding the corresponding values from the fifth replicate. The target zone consisted of bin pairs where both bins were located within the 20th to 40th percentile range of chromosome length, with 0% representing the start of the chromosome and 100% representing the end.

This simulation setting was designed to obtain a controlled differential area in the matrix approximately mimicking a structure similar to a TAD. In particular, this setting should favor 2D-aware tools, e.g. tools that exploit the spatial autocorrelation of the 2D Hi-C matrix (**FIND**, **Selfish**, and **sslHiC**).

Finally, the four matrices (from two conditions) were processed as described in "Type-I error control (H_0 setting)", distinguishing results for the target zone from the others. The precision-recall (PR) curves based on adjusted *p*-value filtering were then obtained to simultaneously assess the precision (i.e. the ratio of bin pairs in the target zone among bin pairs declared positive) and the recall (i.e. the ratio of bin pairs declared positive among bin pairs in the target zone). Note that the recall is also named power in the framework of statistical tests and that "1− Precision" indicates if the test properly controls the FDR.

Table 3 gives the total number of bin pairs for each chromosome and resolution, as well as the number of bin pairs in the target zone.

CTCF depletion study

To test the tools on a real life use case, we retrieved publicly available data from a CTCF depletion study in post-mitotic mouse cells [59]. This study features a Hi-C chromatin structure profiling of a murine erythroblast cell line under two conditions: either under accute depletion of CTCF through an auxin-inducible degron system (CTCF- condition) or in the control condition without auxin-induced depletion (CTCF+ condition). Hi-C libraries were generated, sequenced, and processed for three biological replicates per condition. We downloaded the six corresponding interaction matrices (GEO Accession GSE168251) and ran all the tools on each autosome independently at the resolution of 100 kb.

Although no precise and exhaustive ground truth exists for such a real case dataset, it is well known that the CTCF protein plays a major role in chromatin loop and TAD boundary formation. As reported in the original study, many structural differences are therefore expected between the two conditions, involving in particular genomic regions with a high density of active CTCF binding sites [59]. In order to assess the biological consistency of the predicted differential interactions, we compared the corresponding genomic positions with those of the active CTCF binding sites that were profiled by ChIP-seq experiments on the same cell line (GEO ACCESSION GSE129997, [60]). More precisely, for each 100 kb bin of the genome, we both computed:

- the number of times this bin was included in a bin pair found significantly different by the tool;
- the number of CTCF active sites (called peaks) present in this bin pair.

The joint distribution of these two quantities was thus obtained, and the Spearman correlation was computed to assess the general biological consistency of each tool's results.

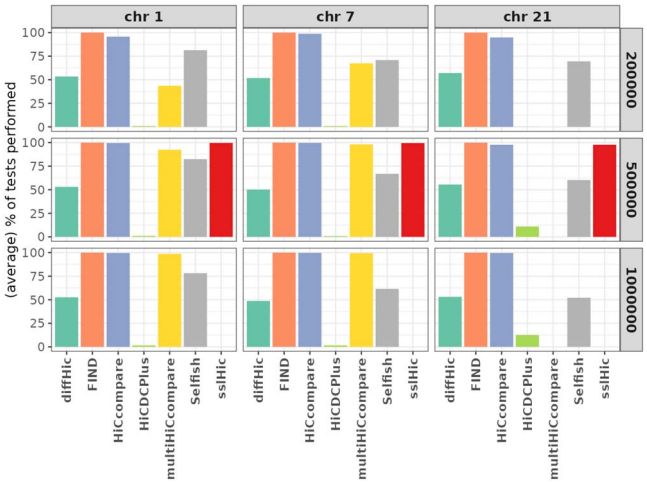


Figure 3. **Average percentage of performed tests** (across the 10 repeats) compared to the number of bin pairs passed as input to the tool (given in Table 3) across chromosomes and resolutions (200kb, 500kb, and 1Mb) in the H_0 setting. **sslHiC** could only be used on 500 kb resolution data.

Computational time

All tools were tested on the same infrastructure (Genotoul-Bioinfo cluster) on a single CPU node, except for **sslHiC** that was tested on a different node because it required GPU. For comparison purposes, we ran the tools on one processor only and recorded computational times in the H_1 setting and for the CTCF depletion study.

Processed data as used in the numerical experiments along with scripts implementing the different tools and performing the result analysis are made available at https://forgemia.inra.fr/scales/replication-chrocodiff.

Results

Number of tested bin pairs

We used the H_0 setting to assess the differences in the number of bin pairs filtered before the test procedure by the different tools. Figure 3 provides the proportion of tests performed for each tool in the H_0 setting (relative to the maximum number of possible tests, as given in Table 3 for each chromosome and resolution). The difference in the numbers of tested bins is thus only due to differences in the filtering step.

The different tools apply pre-filtering steps that resulted in a very different number of tested bin pairs. **HiCcompare** performed a number of tests that is constantly close to the maximum and

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf074/8051526 by guest on 07 May 2026

HiCDCPlus constantly performed a very low number of tests because it only tests (the union of) regions with an interaction considered significantly above the interaction background (FDR adjusted p-value < 10%). For the relatively short chromosome 21, **multiHiCcompare** did not perform any test (all interactions were filtered out at preprocessing). However, it performed a number of tests close the maximum for the other two chromosomes at resolutions 500 kb and 1 Mb. At a 500 kb resolution (the only resolution available for this tool), **sslHiC** performed a number of tests close to the maximum for the three chromosomes. Finally, **diffHic** and **Selfish** filtered out approximately half of the bin pairs.

Note that the differences in the number of tested bin pairs are partially due to default values set differently by different tools for the same parameter: For instance, **HiCcompare** filters out bin pairs with an average A value smaller than the 10th percentile of A values while **multiHiCcompare** filters out bin pairs with an average A value smaller than 5.

Type-I error control (H_0 setting)

Figure 4 provides the percentage of tests declared significant for all chromosomes, resolutions, and tools, based on a 5% and a 1% thresholding of p-values and adjusted p-values. Figure S1 in Supplementary material additionally provides the same plots for a 10% threshold.

In H_0 settings, the percentage of p-values below 5% is expected to be at most 5% if the test is properly calibrated (type-I error control). A percentage much smaller than 5% indicates that the Type-I error control is valid but that the tool is conservative, suggesting that the test may be underpowered in non-H_0 situations. Also, since some tools only returned the adjusted p-value, we also gave the percentage of adjusted p-values below 5%. Since p-values are adjusted to control the FDR, this percentage is expected to be 0 if the test is properly calibrated. However, it is not possible to assess how conservative the test is only based on adjusted p-values.

The results shown in Fig. 4 are remarkably consistent across resolutions. This illustrates the fact that since FDR corresponds to a proportion of false positives, FDR control is *a priori* designed to be comparable across studies with different numbers of tests. Overall, the results show that only **diffHic** and **multiHiCcompare** properly controlled the Type-I error on this dataset, with a percentage of tests declared positive very close to the expected value. Nonetheless, for chromosome 21, **multiHiCcompare** did not perform any test as discussed above (see Fig. 3).

HiCcompare, which does not account for replicates and hence for variability within conditions, suffered from a small excess of false positives (e.g. chromosome 21, 1 Mb resolution, 1% risk). On the contrary, **HiCDCPlus** detected very few false positive results, except for chromosome 21 which displays a massive excess of false positives. This discrepancy between chromosomes could be related to the very low proportion of bin pairs passing the **HiCDCPlus** filters (see Fig. 3). Both **Selfish** and, in particular, **FIND** produced a large number of false positives, as visible in the plots based on adjusted p-values (bottom). For both methods, this could be explained by a statistical issue in the definition of the bin pair-level p-value (lack of multiple testing correction across radii for **Selfish**, and incorrect assumption of independence between aggregated p-values for **FIND**), as explained in Section "Methodological background of the tools." **sslHiC** did not return any positive result based on adjusted p-value thresholding (which is the expected behavior). However, since it does not provide raw p-values, we were not able to assess its proper control of Type-I error. Finally, no large difference was observed between the standard BH correction ("*XXX*-realFDR") and the multiple test correction implemented in **multiHiCcompare** and **HiCcompare**. This is confirmed by the strong linear relationship between these two quantities (Supplementary Fig. S2).

For the tools that returned unadjusted p-values, Fig. 5 provides the ECDF of p-values. Note that the data displayed in Fig. 4 correspond to the values of the ECDF at risk $x = 5\%$ (a) and 1% (b), respectively.

For all resolutions and chromosomes, **diffHic** was the tool closest to the expected uniform distribution, followed closely by the slightly conservative **multiHiCcompare** (Fig. 5). **HiCcompare** exhibited a slight excess of very small p-values; in the area where the p-value is below 0.1%, the ECDF of **HiCcompare** was frequently above the diagonal (see also Fig. 4). This behavior can be explained by its incapacity to account for variability across replicates of one condition, resulting in an excess of false positives. In contrast, **HiCDCPlus** generally displayed the opposite behavior, suggesting a lack of power (especially for chromosomes 1 and 7). However, it occasionally presented a strong excess of false positives, as observed on chromosome 21.

Precision and Recall (H_1 setting)

Figure 6a provides the proportion of tested interactions that are located within the target zone, where positive calls are expected (true signal). This proportion may vary even for tools that rely on similar models or methods, because different data filtering methods are applied before testing for differential interaction. Results confirmed that this filtering step can have *per se* a strong impact on the test. In particular, **HiCDCPlus** predominantly discarded interactions outside the target zone rather than inside, which may be a desirable behavior. However, the overall number of retained interactions was generally very low for this tool (see Fig. 3 and the corresponding discussion). The other tools tended to generally have a proportion of tested interactions in the target zone close to the corresponding proportion in the original matrix (before filtering).

Figure 6b displays PR curves based on predictions computed from adjusted p-values, for all chromosomes, resolutions, and tools. An ideal classifier would have a precision of one and a recall of one. The resulting PR curve, based on various adjusted p-value thresholds, would then be the horizontal line joining the point at $(0, 1)$ coordinates to point at $(1, 1)$ coordinates (i.e. varying power depending on the number of interactions selected by the threshold, but all true positives), followed by a vertical line joining $(1, 1)$ and $(1, 0)$ (i.e. negative interactions are all selected after the positive one, for larger thresholds). The symbols on the PR curves in Fig. 6 b indicate for each tool the obtained precision and recall when thresholding the adjusted p-values provided at risk 5%, a threshold that corresponds to standard practice. A well calibrated tool should have a precision above the dashed horizontal line at 95% (for a clearer visualization of these results, Fig. S3 in Supplementary material also provides the obtained precision and recall for additional adjusted p-values thresholds).

In Fig. 6b, **diffHic** appears to be one of the best tools in the H_1 setting, as it yields curves closest to the ideal classifier in a majority of cases. In particular, it performed best on smaller chromosomes and at higher resolutions. This variability seems to be directly related to the number of performed tests: The smaller the number of tests (chromosome 21 or lower resolutions correspond to smaller numbers of interactions), the better the

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf074/8051526 by guest on 07 May 2026

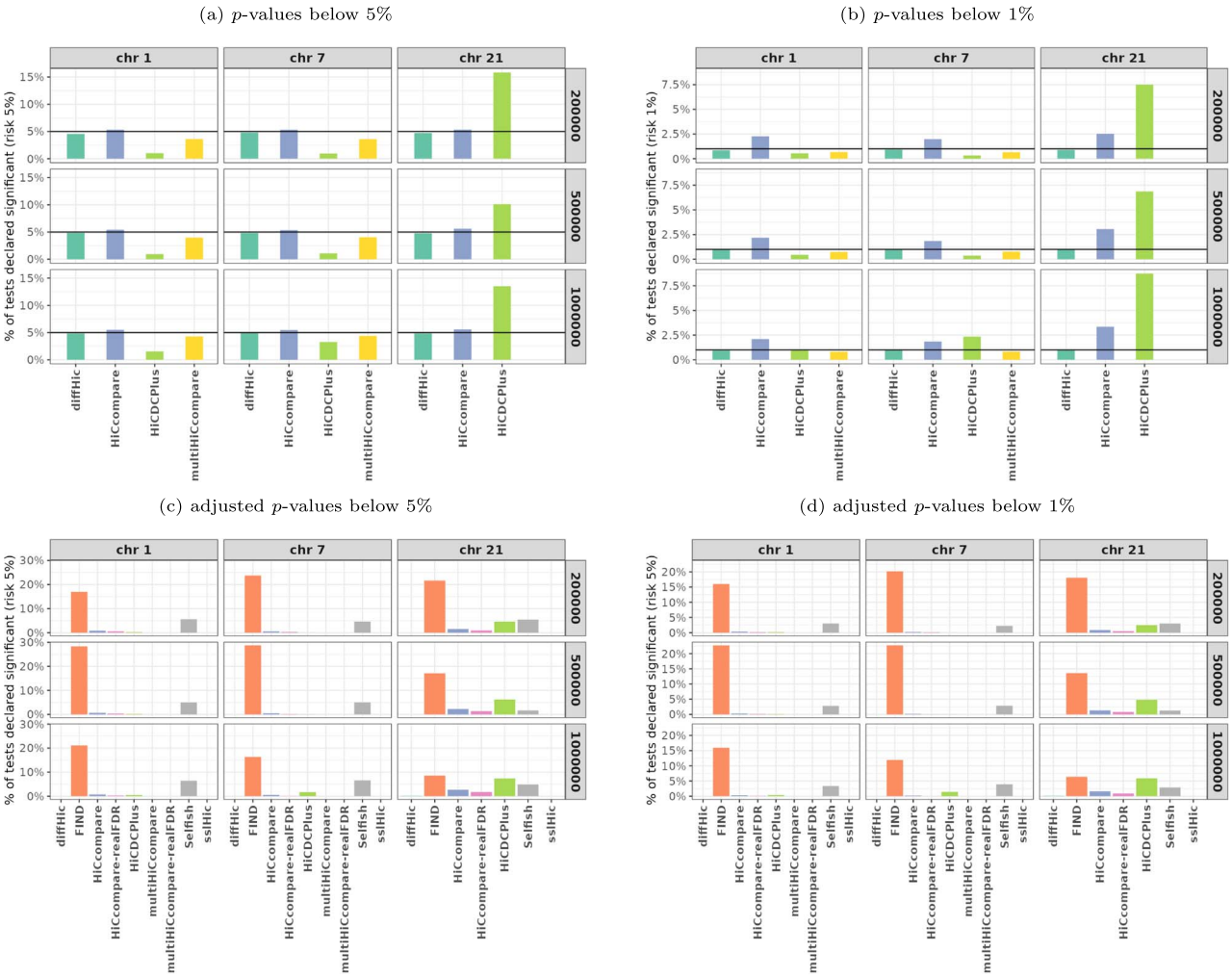


Figure 4. **Percentage of tests declared significant** (H_0 setting) for three chromosomes and three resolutions (200kb, 500kb, and 1Mb). Decisions are taken based on *p*-values (top) and adjusted *p*-values (bottom) with 5% (left) and 1% (right) risks. The black horizontal line (top figures) indicates the risk controlled by raw *p*-values. When the method did not perform a global FDR correction (see Table 1), we re-computed the adjusted *p*-values with the BH method applied to raw *p*-values (when available). This corresponds to columns named "XXX-realFDR" (bottom figures). **sslHiC** could only be used on 500 kb resolution data and **multiHiCcompare** performed no test on chromosome 21 because of its filtering step.

performance. However, in a number of cases, it did not properly control the FDR (the symbol corresponding to the 5% threshold of *p*-values is below the vertical dotted line at 95%). For instance, for chromosome 1, resolution 200 kb, the precision of **diffHic** was between 50 and 75% for the three thresholds.

multiHiCcompare and **HiCcompare** often displayed similar performances (and sometimes better than **diffHic**), with a marked disadvantage of **HiCcompare** for the highest resolution (200 kb). Aside from this resolution, and despite not utilizing information on biological replicates, **HiCcompare** achieved slightly better results than **multiHiCcompare** overall. Note that, similarly to the H_0 setting, the difference between the standard BH correction ("XXX-realFDR") and the multiple testing correction implemented in **multiHiCcompare** and **HiCcompare** is small, with a slight improvement of performances when the standard BH correction is used. However, none of the two tools and the two versions of the correction properly controls the FDR at 5%. The only exceptions are **HiCcompare** at resolutions 1 Mb and 500 kb, but only for chromosomes 1 and 7 and with a recall of zero for chromosome 1.

From a PR curve point of view, **FIND** performed rather well for chromosome 1, especially at resolutions 500 kb and 1 Mb with a curve consistently close or above the ideal classifier. However, its performances were bad for chromosomes 7 and 21, as all interactions had an adjusted *p*-value equal to 1. Note that, even for chromosome 1, **FIND** was far from properly controlling the FDR when thresholding the adjusted *p*-value. In all cases, its precision was close to 0.

HiCDCPlus had a rather heterogeneous and mild performance across chromosomes and resolutions. However, it was systematically the second or third best performing method.

Finally, although being somewhat heterogeneous between chromosomes and resolutions, **Selfish** and **sslHiC** had poor performances on this benchmark, altogether with PR curves usually close or below those of a random classifier.

CTCF depletion study

In order to test the tools in a realistic setting on a full size dataset, we retrieved and analyzed genome-wide Hi-C matrices from a CTCF depletion study in mouse [59] (see Methods).

Figure 7 provides for each tool the joint distribution over the 100 kb bins of the genome between the number of CTCF sites present in the bin (x-axis) and the number of differential interactions in which the bin was involved after comparing matrices

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf074/8051526 by guest on 07 May 2026

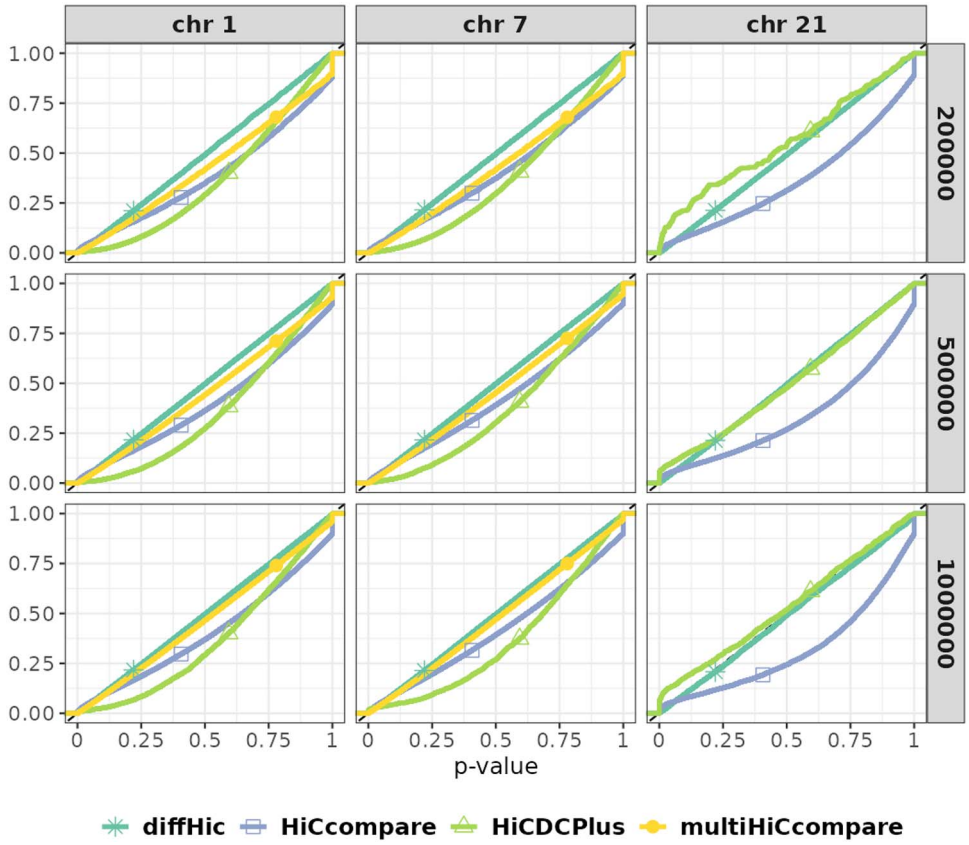


Figure 5. **ECDF of *p*-values** (H_0 setting). Well-calibrated tools are expected to have an ECDF that closely follows the diagonal, corresponding to a uniform distribution of *p*-values under H_0. An ECDF below the diagonal indicates a valid but conservative test, while an ECDF above the diagonal indicates that the test is not properly calibrated, yielding an excess of false positives. **multiHiCcompare** performed no test on chromosome 21 because of its filtering step.

from the CTCF+ and the CTCF- conditions (*y*-axis). Since CTCF depletion is expected to predominantly impact genomic regions with CTCF binding sites, a positive correlation should be observed between these quantities.

This was the case for some of the tools. In particular, Spearman's correlation values of $r = 0.44$, $r = 0.42$, and $r = 0.41$ were, respectively, obtained for **multiHiCcompare**, **HiCDCPlus,** and **diffHic**. Globally, these tools detected more differential interactions between CTCF-rich regions than between CTCF-poor regions, as expected. However, in line with previous results from the H_0 setting (Fig. 3), **HiCDCPlus** realized a low number of tests compared to the total number of bin pairs in the whole dataset. Less than 5% of bin pairs were kept after the filtering step (Supplementary Fig. S4).

On the contrary, no substantial correlation was obtained for the other tools, with $r = 0.04$ for **HiCcompare**, $r = 0.05$ for **Selfish**, and $r = 0.08$ for **FIND**.

Computational time

Figure 8 shows the computational time required for performing the tests in the H_1 setting and for the CTCF depletion dataset. In addition, Supplementary Fig. S5 provides another representation of these results for the H_1 setting with respect to the number of tests performed, and Supplementary Fig. S6 gives the total computational time required for the CTCF depletion dataset.

FIND exhibited large computational times for certain chromosomes and resolutions. However, the scalability of this tool was not the worst: **HiCDCPlus** showed the greatest increase in computational time with respect to the number of performed tests. This can primarily be attributed to the fact that its filtering steps removed most bin pairs, and its computational time scaled with the total number of bin pairs rather than the number of bin pairs remaining after filtering. The tools showing the best scalability were **HiCcompare**, **multiHiCcompare**, and **Selfish**. This result may be partly attributed to the faster implementation of the cyclic LOESS (used for MA and MD normalization) available in **HiCcompare** and **multiHiCcompare**, but not in **diffHic**.

Finally, as **sslHiC** is designed to run on GPU processors, its computational time could not be directly compared with that of the other tools. Despite its relatively short runtime, its computational resource requirements were substantial.

Discussion and conclusion

Our benchmark allowed to evaluate and compare the statistical performances of Hi-C data differential analysis tools on practical examples. Importantly, the results revealed that the FDR was not properly controlled across all tools. This could be due to the small number of samples in our experiments (only two per condition for the H_1 setting), highlighting the importance of that factor. Nonetheless, some tools—particularly **diffHic**—still managed to correctly control the Type-I error rate in the H_0 setting. Additionally, the per-distance-basis FDR correction appeared to have a limited effect, especially when applied to a single chromosome.

Globally, in our benchmark, **diffHic** delivered the best results. It properly controlled the Type-I error rate in the H_0 setting and was the only tool to properly control the FDR in some cases. Its power, for a 5% risk, was also among the best, always larger

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf074/8051526 by guest on 07 May 2026

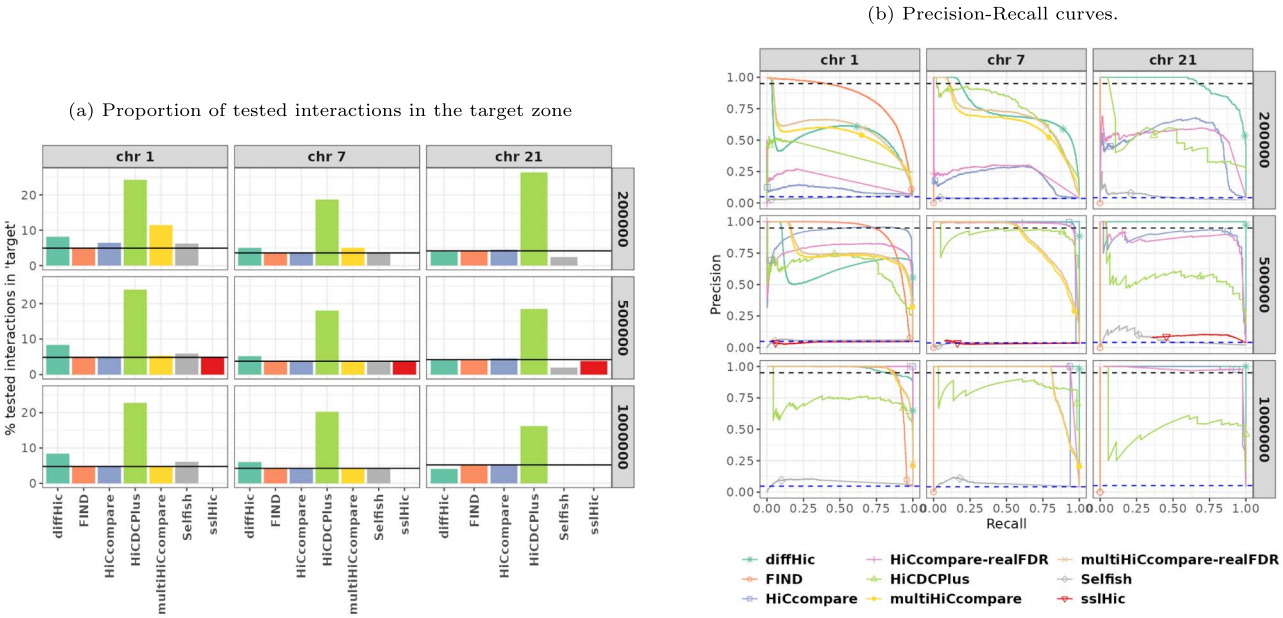


Figure 6. **Results for H_1 setting**. (a) Proportion of tested interactions that are in the target zone for each tool. The horizontal line indicates the proportion of interactions within the target zone of the original data before filtering. The reported proportions (*y*-axis) lie above or below the horizontal line depending on whether the tools (*x*-axis) predominantly filter interactions outside or inside the target zone, respectively. (b) PR curves computed from adjusted *p*-values, displaying recall (or power, *x*-axis) and precision (*y*-axis). For each method, the point corresponding to a threshold = 0.05 (target FDR, as claimed by the method) is marked with a specific symbol. For methods that filtered out all interactions in the target zone before the test, the Recall cannot be computed (because the denominator would be 0). In this case, we arbitrarily represented them with a circle at (0, 0). Finally, the top horizontal dashed line (in black) corresponds to a Precision of 95% and the bottom horizontal dashed line (in blue) corresponds to the precision expectation of a uniform random draw of interactions. **sslHiC** could only be used on 500 kb resolution data and **multiHiCcompare** performed no test on chromosome 21 because of its filtering step.

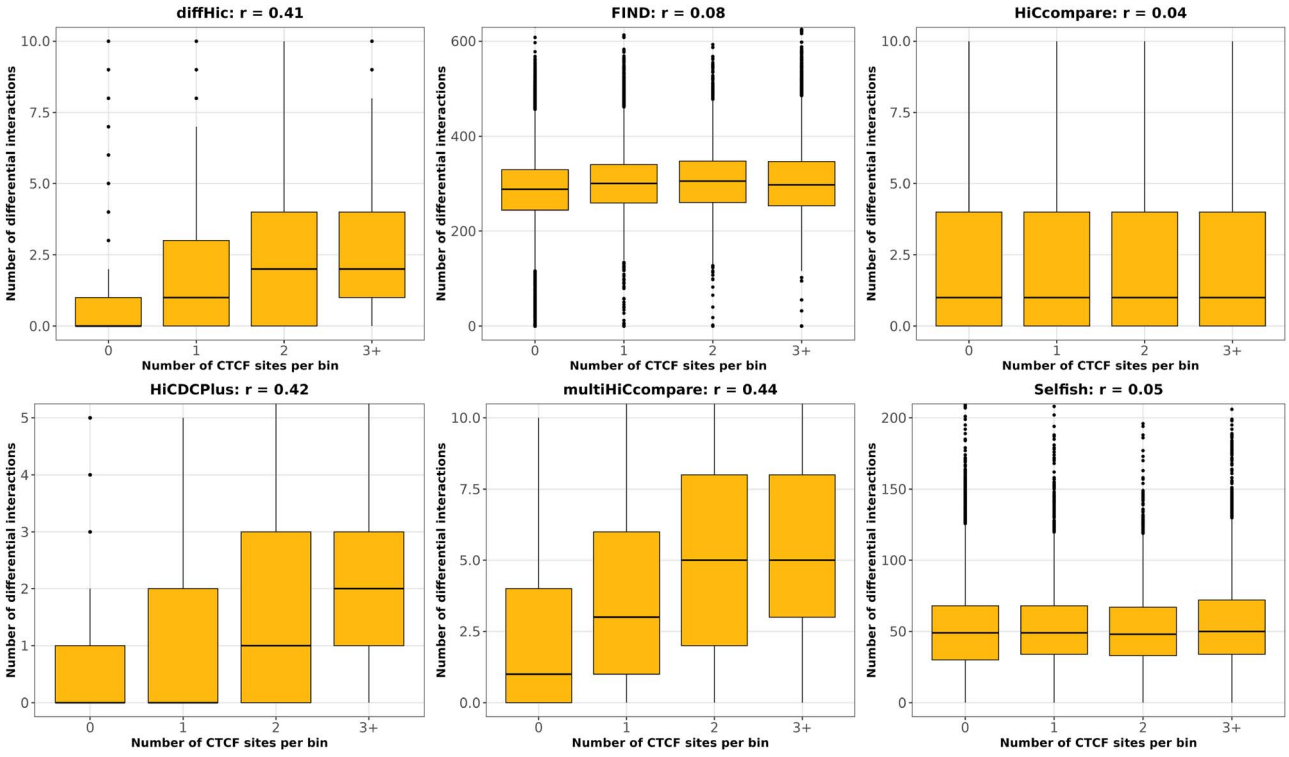


Figure 7. **Joint distribution of the number of CTCF sites and the number of differential interactions** per genomic bin of 100 kb in the CTCF depletion dataset. Each boxplot represents the distribution of differential interactions (*y*-axis) obtained by a given tool (one per panel) between matrices from the CTCF+ and the CTCF- conditions for genomic bins with 0, 1, 2, or at least 3 CTCF sites (*x*-axis). The Spearman correlation between these values across all bins is provided for each tool (*r*). Bins with many CTCF sites are expected to be predominantly involved in differential interactions upon CTCF depletion compared to bins with few CTCF sites, as observed in results from **diffHic** and **HiCDCPlus** for instance (left side).

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf074/8051526 by guest on 07 May 2026

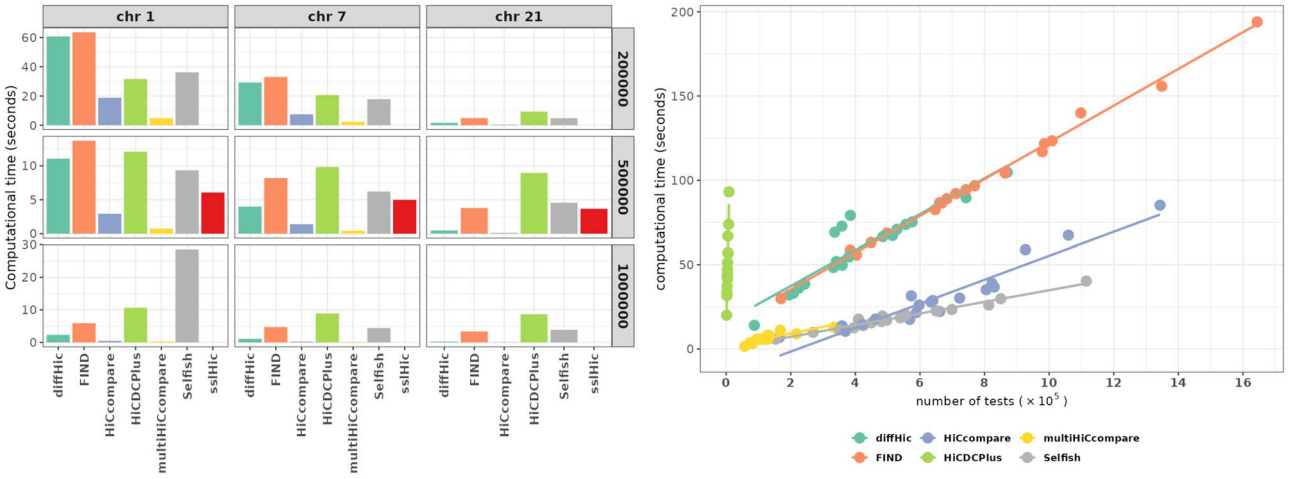


Figure 8. **Computational time.** Left: Computational time in seconds (y-axis) needed for each tool (y-axis) to run in the H_1 setting. **sslHiC** could only be used on 500 kb resolution data and **multiHiCcompare** performed no test on chromosome 21 because of its filtering step. Right: Computational time in seconds (y-axis) versus the number of tests performed in a given chromosome (y-axis) for the CTCF depletion dataset. **sslHiC** could not be used.

than 50% and generally close to 100%. Interestingly, for the lowest resolution, **HiCcompare** also showed interesting performances in the H_1 setting but always exhibited an inflated number of false positives in the H_0 setting and gave disappointing results in the CTCF depletion use case.

diffHic and **multiHiCcompare** produced comparable results in the H_0 setting and, in terms of PR curves, in the H_1 setting. Both tools also showed good biological consistency between Hi-C and ChIP-seq data in the CTCF depletion analysis. This alignment was expected, as they rely on the same model. However, the performance differences observed between the two tools underscore the importance of filtering and the choice of default parameters. Notably, the default filters in **multiHiCcompare** seemed sometimes too stringent (e.g. no results were obtained for some chromosomes in both the simulated and real-world experiments). Additionally, the FPR of **multiHiCcompare** consistently exceeded 30% (and often surpassed 50%) for a 5% risk threshold. The strong impact of preprocessing steps is unsurprising and has been previously acknowledged in other omics studies [61].

HiCDCPlus was also found to be overly stringent in its filtering step, consistently performing a very low number of tests. However, in the CTCF depletion application, it produced good results.

FIND presents an interesting case. In both simulation settings, the tool tended to predict too many false positives for a given threshold. However, on chromosome 1 in the H_1 setting, it demonstrated excellent ordering of interactions based on adjusted *p*-values, with the PR curve closely approaching that of a perfect classifier. This suggests that the adjusted *p*-values returned by **FIND** can serve as a reliable score for ranking interactions by significance level, although they cannot be statistically interpreted. In this case, using higher thresholds than typically expected is recommended. However, for chromosomes 7 and 21, all adjusted *p*-values returned by **FIND** were equal to 1. Overall, contrary to **diffHic**, the performance differences observed with **FIND** do not seem to be directly related to the number of tested interactions. For instance, in the H_1 setting, **FIND** performed better on the largest chromosome but worse for the highest resolution of 200 kb, which has more interactions to test.

Interestingly, 2D-aware tools such as **FIND**, **Selfish**, and **sslHiC** leverage the spatial auto-correlation inherent to the 2D Hi-C matrices in their modeling. However, these tools did not generally show superior performances, even in situations like the H_1 setting, where there is a strong spatial dependency in differential interaction locations within the 2D Hi-C matrix. This suggests that the current methods for incorporating spatial 2D structure may not be effectively capturing its relevance.

Finally, it is worth noting that the currently available tools are still unable to accommodate a wide variety of study designs. Few methods allow to use covariates (see Table 1) and, to the best of our knowledge, no tool is capable of properly handling paired data (e.g. differences between two tissues, with multiple individuals each providing a pair of tissue samples as replicates) or repeated measurement designs (similarly to what is done in mixed models).

In this study, we focused on two datasets, encompassing two simulation settings and a real-world application. Expanding the investigation to include broader datasets and experimental settings would be valuable to assess the robustness of our conclusions across more varied designs, resolutions, and size effects. Additionally, our analysis underscored the need for a deeper understanding of the complex interplay between preprocessing steps—particularly normalization types and filtering—and the models used.

Key Points

- We reviewed and benchmarked available tools for differential analysis of Hi-C matrices.
- Preprocessing steps differed between tools, strongly impacting the results, even for tools with the same type of model.
- None of the tools properly controlled the FDR at the expected rate in our simulation setting. However, some tools effectively controlled the Type-I error in situations where no signal was expected in the data.
- In our simulations, **diffHic** yielded the best overall results. Currently, tools based on a 2D-aware model did not outperform the others.
- Our review highlighted the need for models and tools able to handle paired designs and repeated measurement designs.

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf074/8051526 by guest on 07 May 2026

Acknowledgments

We are grateful to the genotoul bioinformatics platform Toulouse Occitanie (Bioinfo Genotoul, https://doi.org/10.15454/1.5572369328961167E12) for providing assistance, as well as computing and storage resources.

Author contributions

SF, PN, MZ, and NV conceived the project and designed the study. All authors contributed to the review and to the benchmark. All authors wrote, read, revised, and approved the manuscript.

Supplementary data

Supplementary data are available at *Briefings in Bioinformatics* online.

Conflicts of interest: The authors declare no conflict of interest.

Funding

This work is funded by the INRAE/DIGIT-BIO network ChrocoNET and by the CNRS project SCALES (Mission "Osez l'interdiscip linarité"). E.J.'s PhD is funded by INRAE.

Data availability

For the H_0 and H_1 settings, raw sequencing data were obtained from the ENCODE project https://www.encodeproject.org/ using accession ENCSR295BDK. Processed Hi-C data (by chromosome, resolution, and technical replicates) and corresponding quality controls are available at https://doi.org/10.57745/LR0W9R.

For the CTCF dataset, Hi-C matrices and ChIP-seq peaks were retrieved from GEO using accession GSE168251 and GSE129997, respectively. Processed Hi-C data (by chromosome at 100 kb resolution) and converter script are available at https://doi.org/10.57745/LR0W9R.

References

1. Bonev B, Cavalli G. Organization and function of the 3D genome. *Nat Rev Genet* 2016;**17**:661–78. https://doi.org/10.1038/nrg.2016.112
2. Lieberman-Aiden E, Van Berkum NL, Williams L. *et al*. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* 2009;**326**:289–93. https://doi.org/10.1126/science.1181369
3. Rao SSP, Huntley MH, Durand NC. *et al*. A 3D map of the human genome at kilobase resolution reveals principle of chromatin looping. *Cell* 2014;**159**:1665–80. https://doi.org/10.1016/j.cell.2014.11.021
4. Dali R, Blanchette M. A critical assessment of topologically associating domain prediction tools. *Nucleic Acid Res* 2017;**45**:2994–3005. https://doi.org/10.1093/nar/gkx145
5. Zufferey M, Tavernari D, Oricchio E. *et al*. Comparison of computational methods for the identification of topologically associating domains. *Genome Biol* 2018;**19**:217. https://doi.org/10.1186/s13059-018-1596-9
6. Liu L, Han K, Huimin Sun L. *et al*. A comprehensive review of bioinformatics tools for chromatin loop calling. *Brief Bioinform* 2023;**24**:bbad072. https://doi.org/10.1093/bib/bbad072
7. Lupiáñez DG, Kraft K, Heinrich V. *et al*. Disruptions of topological chromatin domains cause pathogenic rewiring of gene-enhancer interactions. *Cell* 2015;**161**:1012–25. https://doi.org/10.1016/j.cell.2015.04.004
8. Spielmann M, Lupiáñez DG, Mundlos S. Structural variation in the 3D genome. *Nat Rev Genet* 2018;**19**:453–67. https://doi.org/10.1038/s41576-018-0007-0
9. Marieke Oudelaar A, Higgs DR. The relationship between genome structure and function. *Nat Rev Genet* 2020;**22**:154–68. https://doi.org/10.1038/s41576-020-00303-x
10. Gunsalus LM, McArthur E, Gjoni K. *et al*. Comparing chromatin contact maps at scale: methods and insights bioRxiv preprint 2023.04.04.535480. 2023.
11. Yang T, Zhang F, Yardimci GG. *et al*. HiCRep: assessing the reproducibility of Hi-C data using a stratum-adjusted correlation coefficient. *Genome Res* 2017;**27**:1939–49. https://doi.org/10.1101/gr.220640.117
12. Wang Z, Zhang Y, Zang C. BART3D: inferring transcriptional regulators associated with differential chromatin interactions from Hi-C data. *Bioinformatics* 2021;**37**:3075–8. https://doi.org/10.1093/bioinformatics/btab173
13. Soler-Vila P, Cuscó P, Farabella I. *et al*. Hierarchical chromatin organization detected by TADpole. *Nucleic Acids Res* 2020;**48**:e39. https://doi.org/10.1093/nar/gkaa087
14. Mourad R. TADreg: a versatile regression framework for TAD identification, differential analysis and rearranged 3D genome prediction. *BMC Bioinformatics* 2022;**23**:82. https://doi.org/10.1186/s12859-022-04614-0
15. Chen F, Li G, Zhang MQ. *et al*. HiCDB: a sensitive and robust method for detecting contact domain boundaries. *Nucleic Acids Res* 2018;**46**:11239–50. https://doi.org/10.1093/nar/gky789
16. Chakraborty A, Wang JG, Ay F. dcHiC detects differential compartments across multiple Hi-C datasets. *Nat Commun* 2022;**13**:6827. https://doi.org/10.1038/s41467-022-34626-6
17. Hua D, Ming G, Zhang X. *et al*. DiffDomain enables identification of structurally reorganized topologically associating domains. *Nat Commun* 2024;**15**:502. https://doi.org/10.1038/s41467-024-44782-6
18. Cook KB, Hristov BH, Le Roch KG. *et al*. Measuring significant changes in chromatin conformation with ACCOST. *Nucleic Acids Res* 2020;**48**:2303–11. https://doi.org/10.1093/nar/gkaa069
19. Lun ATL, Smyth GK. diffHic: a Bioconductor package to detect differential genomic interactions in Hi-C data. *BMC Bioinformatics* 2015;**16**:258. https://doi.org/10.1186/s12859-015-0683-0
20. Djekidel MN, Chen Y, Zhang MQ. FIND: difFerential chromatin INteractions Detection using a spatial Poisson process. *Genome Res* 2018;**28**:412–22. https://doi.org/10.1101/gr.212241.116
21. Stansfield JC, Cresswell KG, Vladimirov VI. *et al*. HiCcompare: an R-package for joint normalization and comparison of HI-C datasets. *BMC Bioinformatics* 2018;**19**:279. https://doi.org/10.1186/s12859-018-2288-x
22. Sahin M, Wong W, Zhan Y. *et al*. HiC-DC+ enables systematic 3D interaction calls and differential analysis for Hi-C and HiChIP. *Nat Commun* 2021;**12**:3366. https://doi.org/10.1038/s41467-021-23749-x
23. Stansfield JC, Cresswell KG, Dozmorov MG. multiHiCcompare: joint normalization and comparative analysis of complex Hi-C experiments. *Bioinformatics* 2019;**35**:2916–23. https://doi.org/10.1093/bioinformatics/btz048
24. Ardakany AR, Ay F, Lonardi S. Selfish: discovery of differential chromatin interactions via a self-similarity measure. *Bioinformatics* 2019;**35**:i145–53. https://doi.org/10.1093/bioinformatics/btz362

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf074/8051526 by guest on 07 May 2026

25. Li H, He X, Kurowski L. *et al*. Improving comparative analyses of Hi-C data via contrastive self-supervised learning. *Brief Bioinform* 2023;**24**:bbad193. https://doi.org/10.1093/bib/bbad193
26. Nicoletti C. Methods for the differential analysis of Hi-C data. In: Bicciato S, Ferrari F (eds.), *Hi-C Data Analysis, Volume 2301 of Methods in Molecular Biology*, pp. 61–95. New York, NY, USA: Humana Press,Springer, 202210.1007/978-1-0716-1390-0_4.
27. Galan S, Machnik N, Kruse K. *et al*. CHESS enables quantitative comparison of chromatin contact data and automatic feature extraction. *Nat Genet* 2020;**52**:1247–55. https://doi.org/10.1038/s41588-020-00712-y
28. Bourgon R, Gentleman R, Huber W. Independent filtering increases detection power for high-throughput experiments. *Proc Natl Acad Sci* 2010;**107**:9546–51. https://doi.org/10.1073/pnas.0914005107
29. Rau A, Gallopin M, Celeux G. *et al*. Data-based filtering for replicated high-throughput transcriptome sequencing experiments. *Bioinformatics* 2013;**29**:2146–52. https://doi.org/10.1093/bioinformatics/btt350
30. Lyu H, Liu E, Zhifang W. Comparison of normalization methods for Hi-C data. *Biotechniques* 2019;**68**:56–64. https://doi.org/10.2144/btn-2019-0105
31. Robinson MD, Oshlack A. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol* 2010;**11**:R25. https://doi.org/10.1186/gb-2010-11-3-r25
32. Ballman KV, Grill DE, Oberg AL. *et al*. Faster cyclic loess: normalizing RNA arrays via linear models. *Bioinformatics* 2004;**20**:2778–86. https://doi.org/10.1093/bioinformatics/bth327
33. Lun ATL, Smyth GK. Csaw: a bioconductor package for differential binding analysis of ChIP-seq data using sliding windows. *Nucleic Acids Res* 2016;**44**:e45. https://doi.org/10.1093/nar/gkv1191
34. Ay F, Noble WS. Analysis methods for studying the 3D architecture of the genome. *Genome Biol* 2015;**16**:183. https://doi.org/10.1186/s13059-015-0745-7
35. Imakaev M, Fudenberg G, McCord RP. *et al*. Iterative correction of Hi-C data reveals hallmarks of chromosome organization. *Nat Methods* 2012;**9**:999–1003. https://doi.org/10.1038/nmeth.2148
36. Knight PA, Ruiz D. A fast algorithm for matrix balancing. *IMA J Numer Anal* 2013;**33**:1029–47. https://doi.org/10.1093/imanum/drs019
37. Durand NC, Robinson JT, Shamim MS. *et al*. Juicebox provides a visualization system for Hi-C contact maps with unlimited zoom. *Cell Systems* 2016;**3**:99–101. https://doi.org/10.1016/j.cels.2015.07.012
38. Servant N, Varoquaux N, Heard E. *et al*. Effective normalization for copy number variation in Hi-C data. *BMC Bioinformatics* 2018;**19**:313. https://doi.org/10.1186/s12859-018-2256-5
39. Bacciu D, Errica F, Micheli A. *et al*. A gentle introduction to deep learning for graphs. *Neural Netw* 2020;**129**:203–21. https://doi.org/10.1016/j.neunet.2020.06.006
40. Zhou J, Cui G, Shengding H. *et al*. Graph neural networks: a review of methods and applications. *AI Open* 2020;**1**:57–81. https://doi.org/10.1016/j.aiopen.2021.01.001
41. Robinson MD, McCarthy DJ, Smyth GK. edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 2010;**26**:139–40. https://doi.org/10.1093/bioinformatics/btp616
42. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol* 2014;**15**:550. https://doi.org/10.1186/s13059-014-0550-8
43. Song C, Tseng GC. Hypothesis setting and order statistic for robust genomic meta-analysis. *Ann Appl Stat* 2014;**8**:777–800. https://doi.org/10.1214/13-AOAS683
44. Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J R Stat Soc Series B* 1995;**57**:289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x
45. Abdennur N, Mirny LA. Cooler: scalable storage for Hi-C data and other genomically labeled arrays. *Bioinformatics* 2019;**36**:311–6.
46. Kruse K, Hug CB, Vaquerizas JM. FAN-C: a feature-rich framework for the analysis and visualisation of chromosome conformation capture data. *Genome Biol* 2020;**21**:303. https://doi.org/10.1186/s13059-020-02215-9
47. Servant N, Varoquaux N, Lajoie BR. *et al*. HiC-pro: an optimized and flexible pipeline for Hi-C data processing. *Genome Biol* 2015;**16**:259. https://doi.org/10.1186/s13059-015-0831-x
48. Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* 2010;**26**:841–2. https://doi.org/10.1093/bioinformatics/btq033
49. Lun ATL, Perry M, Ing-Simmons E. Revised infrastructure for genomic interactions: bioconductor classes for Hi-C, ChIA-PET and related experiments. *F1000Res* 2016;**5**:950. https://doi.org/10.12688/f1000research.8759.1
50. Huber W, Carey VJ, Gentleman R. *et al*. Orchestrating high-throughput genomic analysis with Bioconductor. *Nat Methods* 2015;**12**:115–21. https://doi.org/10.1038/nmeth.3252
51. Heinz S, Texari L, Hayes MGB. *et al*. Transcription elongation can affect genome 3D structure. *Cell* 2018;**174**:1522–1536.e22. https://doi.org/10.1016/j.cell.2018.07.047
52. Dìaz N, Kruse K, Erdmann T. *et al*. Chromatin conformation analysis of primary patient tissue using a low input Hi-C method. *Nat Commun* 2018;**9**:4938. https://doi.org/10.1038/s41467-018-06961-0
53. Rao SSP, Su-Chen H, Hilaire BGS. *et al*. Cohesin loss eliminates all loop domains. *Cell* 2017;**171**:305–320.e24. https://doi.org/10.1016/j.cell.2017.09.026
54. Benidt S, Nettleton D. SimSeq: a nonparametric approach to simulation of RNA-sequence datasets. *Bioinformatics* 2015;**31**:2131–40. https://doi.org/10.1093/bioinformatics/btv124
55. Chalise P, Raghavan R, Fridley B. *InterSIM: Simulation of Inter-Related Genomic Datasets* 2018. R package (version 2.2.0).
56. Gerard D. Data-based RNA-seq simulations by binomial thinning. *BMC Bioinformatics* 2020;**21**:206. https://doi.org/10.1186/s12859-020-3450-9
57. The ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature* 2012;**489**:57–74. https://doi.org/10.1038/nature11247
58. Ewels PA, Peltzer A, Fillinger S. *et al*. The nf-core framework for community-curated bioinformatics pipelines. *Nat Biotechnol* 2020;**38**:276–8. https://doi.org/10.1038/s41587-020-0439-x
59. Zhang H, Lam J, Zhang D. *et al*. CTCF and transcription influence chromatin structure re-configuration after mitosis. *Nat Commun* 2021;**12**:5157. https://doi.org/10.1038/s41467-021-25418-5
60. Zhang H, Emerson DJ, Gilgenast TG. *et al*. Chromatin structure dynamics during the mitosis-to-G1 phase transition. *Nature* 2019;**576**:158–62. https://doi.org/10.1038/s41586-019-1778-y
61. Luppi AI, Gellersen HM, Liu Z-Q. *et al*. Systematic evaluation of fMRI data-processing pipelines for consistent functional connectomics. *Nat Commun* 2024;**15**:4745. https://doi.org/10.1038/s41467-024-48781-5

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf074/8051526 by guest on 07 May 2026

© The Author(s) 2025. Published by Oxford University Press. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.
Briefings in Bioinformatics, 2025, **26(2)**, bbaf074
https://doi.org/10.1093/bib/bbaf074
Review

Chapitre 3

hicream : méthode d'identification de régions différentielles dans les données Hi-C

Ce chapitre contient l'article Jorge et al. [2026] publié dans *Bioinformatics*. Les annexes de ce chapitre se trouvent en fin de manuscrit, en annexe B. Un résumé de l'article en français est donné ci-dessous.

Motivation. L'organisation tridimensionnnelle (3D) du génome joue un rôle central dans de nombreux processus biologiques et la perturber peut avoir un impact critique sur les fonctions cellulaires. Par conséquent, il est crucial de détecter de tels changements et cela peut être fait grâce à l'analyse différentielle de données Hi-C. À travers la comparaison de données Hi-C entre différentes conditions biologiques, l'analyse différentielle de données Hi-C a pour but d'identifier des changements dans l'organisation de la chromatine qui soient statistiquement significatifs et pertinents d'un point de vue biologique. Cependant, la plupart des méthodes existantes se concentrent soit sur un unique type de structure défini a priori (TADs, boucles, ou compartiments), limitant de fait leur capacité à découvrir de nouvelles variations, soit produisent des résultats fortement localisés et disparates qui sont difficiles à interpréter en terme d'organisation génomique d'un niveau supérieur.

Résultats. Nous introduisons hicream, un nouveau cadre d'analyse différentielle de données Hi-C qui identifie des régions de forme arbitraire, permettant ainsi de découvrir de nouvelles structures différentielles. En combinant l'analyse différentielle basée sur les pixels et un clustering guidé par les données pour définir des régions candidates de la carte Hi-C, notre approche fournit une mesure interprétable du signal différentiel pour chaque région grâce à une stratégie d'inférence *post hoc*. Les régions différentielles qui en résultent peuvent être explorées grâce à une interface de visualisation interactive.

Implémentation et disponibilité Le package hicream est disponible sur `https://cran.r-project.org/package=hicream`. Les données utilisées dans cet article sont disponibles à `https://doi.org/10.57745/IEUSLV` et les scripts permettant de reproduire

les résultats sont disponibles à https://forge.inrae.fr/scales/replication-repository-for-hicream-article.

Bioinformatics, 2026, **42**(9), btag631
https://doi.org/10.1093/bioinformatics/btag631
Published: 25 August 2026
Original Paper | Genome analysis

OXFORD

hicream: A flexible framework to identify significantly different regions in Hi-C data

Elise Jorge[1,*], Toby Dylan Hocking[2], Pierre Neuvial[3], Nathalie Vialaneix[4], Sylvain Foissac[1,*]

[1]GenPhySE, Univ Toulouse, Toulouse INP, ENVT, INRAE, Castanet-Tolosan 31326, France
[2]LASSO lab, Département d'informatique, Université de Sherbrooke, Sherbrooke QC J1K-2R1, Canada
[3]Institut de Mathématiques de Toulouse, UMR 5219, Université de Toulouse, CNRS UPS, Toulouse 31062, France
[4]Université Fédérale de Toulouse, INRAE, MIAT, Castanet-Tolosan 31326, France
*Corresponding authors. Elise Jorge, GenPhySE, Univ Toulouse, Toulouse INP, ENVT, INRAE, Castanet-Tolosan 31326, France. E-mail: elise.jorge@inrae.fr; Sylvain Foissac, GenPhySE, Univ Toulouse, Toulouse INP, ENVT, INRAE, Castanet-Tolosan 31326, France. E-mail: sylvain.foissac@inrae.fr
Associate Editor: Can Alkan

Abstract

Motivation: **The three-dimensional (3D) organization of the genome plays a central role in many biological processes, and its disruption can critically impair cellular function. Detecting such changes is therefore crucial and can be achieved through differential analysis of Hi-C data. By comparing Hi-C data across biological conditions, differential Hi-C analysis aims to identify statistically significant and biologically relevant changes in chromatin organization. However, most existing methods either target a single predefined type of structure (e.g. TADs, loops, or compartments), thereby limiting their ability to uncover novel or complex structural variations, or produce highly local and disconnected results that are difficult to interpret in terms of higher-order genome organization.**

Results: **We introduce *hicream*, a novel framework for differential Hi-C analysis that identifies regions of arbitrary shape, allowing to uncover new differential structures. By combining pixel-level differential analysis and data-driven clustering to define candidate regions of the Hi-C interaction matrix, our approach provides an interpretable measure of the differential signal for each region through a post hoc inference strategy. The resulting differential regions can be explored using an interactive visualization interface.**

Availability and Implementation: hicream **is available at https://cran.r-project.org/package=hicream**

1 Introduction

In eukaryotic cells, chromosomes are compacted within the nucleus and organized into three-dimensional (3D) structures at different levels, including chromosomal territories, chromatin compartments, topologically associating domains (TADs), and chromatin loops. This three-dimensional (3D) organization regulates gene expression and plays a crucial role in various biological processes such as cell division and differentiation (Oudelaar *et al.* 2020). It has been shown to be involved in cancers and congenital anomalies (Lupiáñez *et al*. 2015, Spielmann *et al*. 2018).

The 3D genome architecture can be profiled using Hi-C (Lieberman-Aiden *et al*. 2009), which quantifies interaction frequencies between genomic intervals ("bins") along the chromatin. Hi-C data are commonly represented as sparse, symmetric interaction matrices, where each entry records the number of detected interactions between bins i and j, where (i, j) is called a "pixel." By comparing such matrices across biological conditions, differential Hi-C analysis aims at identifying statistically significant and biologically relevant changes in chromatin organization (Gjoni *et al*. 2025).

A first class of approaches to differential analysis of Hi-C data focuses on structures and aims to identify differential regions of the Hi-C matrix corresponding to an *a priori* defined type of structure, such as chromatin loops (Lareau and Aryee 2018), TADs (Mourad 2022, Hua *et al*. 2024) or A/B compartments (Chakraborty *et al*. 2022, Maigné *et al*. 2025). This strategy constrains the analysis to one type of structure at a time, requiring to use multiple tools to capture variations across different levels of chromatin organization. Moreover, it either relies on rigid, structure-specific shape assumptions for differential regions, or is limited to detecting differential boundary locations. This leaves aside the diversity of shapes that structural variations can produce: For instance, Cresswell and Dozmorov (2020) identify several types of TAD changes with distinct differential region shapes, such as a triangle for TAD strengthening and a stripe for

Received: 23 February 2026. **Revised:** 15 July 2026. **Accepted:** 4 August 2026
© The Author(s) 2026. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Downloaded from academic.oup.com/bioinformatics/article/42/9/btag631/8770358 by guest on 24 September 2026

a boundary shift. Directly identifying differential regions in a shape-agnostic manner is therefore more flexible and comprehensive. Last, analyzing each structural type independently does not naturally account for complex chromatin variations arising from the interplay of multiple structural elements across different scales of organization.

A second class of approaches, referred to as "pixel-based" differential analysis, focuses on identifying differential pixels individually (Jorge *et al*. 2025b). For each pixel (i,j), a test is performed and a *P*-value is computed. A global multiple testing correction using the Benjamini-Hochberg (BH) procedure (Benjamini and Hochberg 1995) is then applied across all pixels to control the False Discovery Rate (FDR), defined as the expected proportion of false positives among the rejected hypotheses. The main limitation of this approach lies in the disconnected nature of the results and its dependence on matrix resolution. Restricting the analysis to independent pixels is not well suited to detecting structural variations that extend beyond regions of an arbitrary pixel size. To address this limitation, the *diffHic* package (Lun and Smyth 2015) introduced the *diffHic clustering* method, which aims to form clusters of significant pixels based on adjusted *P*-values. However, as noted in the package documentation, this procedure does not provide formal statistical guarantees, as it relies on a potentially biased estimate of the cluster-level FDR.

To overcome the limitations of these approaches, we introduce *hicream*, a data-driven framework for detecting genomic structural variations by identifying differential regions of arbitrary size and shape between groups of Hi-C matrices. *hicream* builds upon a pixel-based differential analysis followed by a clustering step. It then combines the resulting clusters and associated *P*-values with a *post hoc* statistical inference framework (Goeman and Solari 2011) to estimate the proportion of differential pixels within each cluster and to select clusters with the highest estimated proportions. We evaluated *hicream*, compared it against the *diffHic clustering* approach, and applied it to two independent real datasets, identifying differential regions. Results show consistency with external data, including active CTCF binding sites from ChIP-seq experiments and enriched transcription factor binding sites (TFBS).

2 Materials and methods

2.1 Differential analysis of Hi-C data

In this article, we address the comparison between two groups of $r = r_1 + r_2$ Hi-C matrices, which are $(p \times p)$-symmetric matrices, with nonnegative integer entries for p bins. These matrices are denoted $M^{g,l}$, where $g \in \{1,2\}$ stands for the group to which the matrix belongs (also called the "condition") and $l \in \{1,\dots,r_g\}$ corresponds to the l-th replicate of condition g. Note that these replicates are not assumed to be paired between conditions. The matrix entries are denoted by $m_{ij}^{g,l}$. They represent the interaction frequency (in terms of number of reads) between bins i and j for the l-th matrix of condition g. Note that we only consider intra-chromosomal interactions, *i.e*, matrices corresponding to a single chromosome.

In this setting, the objective of Hi-C data differential analysis is to identify changes within the 3D conformation of the genome between the two conditions $g = 1$ and $g = 2$, i.e. differential clusters of pixels.

2.2 Proposed workflow: hicream

Our approach is implemented in the R package *hicream*, following the workflow illustrated in Fig. 1.

A. A statistical test is performed for each pixel to assess significant differences between conditions, resulting in a *p*-value map.

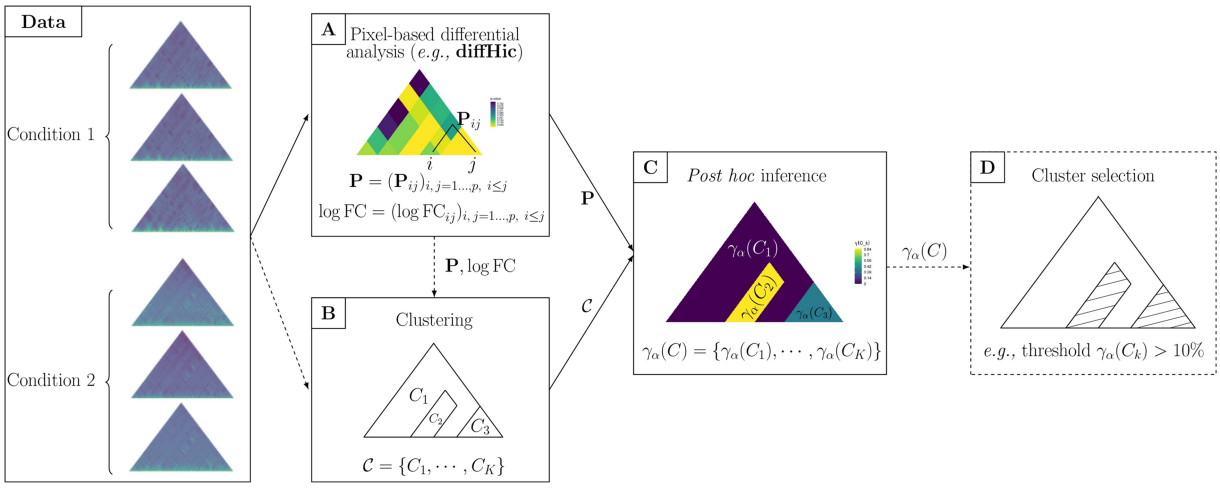


Figure 1 The hicream workflow. Input data correspond to Hi-C matrices from two different biological conditions, with several replicates in each. (A) A pixel-based differential analysis is performed (e.g. using *diffHic*), rendering a map of *P*-values where each *P*-value $\mathbf{P}_{ij}$ corresponds to the test performed for pixel (i,j). (B) A clustering of the pixels is then obtained, providing K clusters of pixels $\mathcal{C} = \{C_1,\dots,C_K\}$. The clustering step can take as inputs either the Hi-C matrices—regardless of their condition—or the results of the pixel-level differential analysis. (C) The *post hoc* inference step takes as inputs the *P*-values computed at step (A) and the clusters of pixels defined at step (B) to compute $\gamma_\alpha(C_k)$, the (minimal) TDP (True Discovery Proportion) for each cluster C_k. (D) An optional step of cluster selection can finally be performed by thresholding $\gamma_\alpha(C_k)$.

Downloaded from academic.oup.com/bioinformatics/article/42/9/btag631/8770358 by guest on 24 September 2026

We chose *diffHic* to perform this differential analysis, as explained in Section 2.2.1;

B. A set of pixel clusters, $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$, is either obtained by an automated clustering method, or chosen by the user. Possible clustering choices are discussed in Section 2.2.2, including data-driven methods;
C. The *p*-value map and the clusters are combined using a *post hoc* approach described in Section 2.2.3 to provide a lower bound on the true discovery proportion (TDP) in each cluster of $\mathcal{C}$, $(\gamma_\alpha(C_k))_{k=1,\ldots,K}$, with high probability $1-\alpha$;
D. (optional) The user can define a TDP threshold to select some of these clusters (those with enough statistical evidence) for further interpretation (e.g. enrichment analyses).

These steps are described in the next sections and further information on the method is also provided in Supplementary Section S1.

2.2.1 Pixel-based differential analysis

Our recent benchmark of pixel-based differential analysis methods (Jorge *et al.* 2025b) highlighted the wide diversity of available approaches and their heterogeneous statistical performance. Among them, the method implemented in the R/Bioconductor package *diffHic* (Lun and Smyth 2015) showed strong results and was therefore selected as the default option in *hicream*. For every pixel (i,j), it results in a *P*-value half-matrix, $(\mathbf{P}_{ij})_{i,j=1\ldots,p,i\leq j}$, which contains the result of $m \leq \frac{p(p+1)}{2}$ tests performed at the pixel level, where m is the number of pixels for which at least one interaction has been observed in one of the r Hi-C matrices.

2.2.2 Pixel clustering

The second step of the workflow consists of defining pixel clusters, $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$. Importantly, the *hicream* workflow can cope with any number of pixel clusters, including overlapping ones, without compromising its statistical guarantees. Users may therefore define regions based on prior knowledge or specific assumptions.

In the absence of such prior knowledge or expectations regarding regions of interest, *hicream* proposes an automatic, data-driven clustering method. This strategy identifies clusters of pixels that are similar either in terms of interaction counts (*hicream counts*) or with respect to the results of the differential analysis (*hicream diff*), while respecting the 2D organization of the Hi-C matrix. Both approaches rely on a constrained hierarchical clustering based on Ward's linkage (Randriamihamison *et al.* 2021). This method takes as input a matrix $\mathbf{X}$ with m rows. It starts with an initial clustering where each pixel is a cluster and greedily agglomerates pairs of adjacent clusters that provide the smallest change in within-cluster inertia.

In *hicream counts*, for each of the m pixels $q := (i,j)$, $\mathbf{X}$ contains its (normalized by *diffHiC*) counts across replicates and conditions: $\mathbf{X}_{q,.} = (m_{ij}^{1,1}, \ldots, m_{ij}^{1,r_1}, m_{ij}^{2,1}, \ldots, m_{ij}^{2,r_2})$. In *hicream diff*, $\mathbf{X}$ uses the results of the differential analysis, aggregating the log-transformed adjusted *P*-values and the log-fold change values: $\mathbf{X}_{q,.} = (\log \mathbf{P}_{ij}^{\text{adj}}, \text{logFC}_{ij})$. The spatial dependence between adjacent pixels from the 2D Hi-C matrix is encoded in the connectivity graph $\mathcal{G}$, where nodes V correspond to pixels and edges link adjacent pixels: Two pixels (i_1,j_1) and (i_2,j_2) are said to be adjacent if $(i_2,j_2) = (i_1,j_1 \pm 1)$ or $(i_2,j_2) = (i_1 \pm 1, j_1)$. Further details on how missing data are handled in the connectivity graph are provided in Supplementary Section S1.

This approach gives a hierarchical clustering that can be represented as a dendrogram and is a collection of clusterings. The final clustering is selected by finding the elbow point of the Ward's linkage.

In practice, we used the algorithm implemented in *scikit-learn* (Pedregosa *et al.* 2011) which is a greedy algorithm, iteratively merging two clusters while respecting the adjacency constraint. The final clustering was identified using the Python package *kneebow* that implements the elbow criterion. Other clustering strategies and model selection criteria could also be used within our framework.

2.2.3 *Post hoc* inference

Post hoc inference methods (Goeman and Solari 2011) provide statistical guarantees for arbitrary sets of clusters. For a subset of pixels $C \subset \mathcal{H} := \{1, \ldots, m\}$, let $\text{TDP}(C)$ be the proportion of true positives (or True Discovery Proportion) in C, that is, the (unknown) proportion of pixels (i,j) in C such that the null hypothesis $\text{H}_{0,ij}$ is false. Given a target risk level $\alpha \in (0,1)$, *post hoc* inference builds, from the data, a function γ_α which assigns to each subset $C \subseteq \mathcal{H}$ a value $\gamma_\alpha(C) \in [0,1]$, such that

$$\mathbb{P}\Big(\forall C \subset \mathcal{H},\ \text{TDP}(C) \geq \gamma_\alpha(C)\Big) \geq 1-\alpha. \tag{1}$$

With probability at least $1-\alpha$, for any C, the proportion of true positives in C is at least $\gamma_\alpha(C)$. A function γ_α satisfying (1) is called a *post hoc bound* for the TDP.

While both FDR and *post hoc* inference provide statistical control on the proportion of errors ($\text{FDR} = \mathbb{E}(\text{FDP})$ where $\text{FDP} = 1 - \text{TDP}$ is the proportion of false discoveries), we emphasize that they provide fundamentally different statistical guarantees:

- First, while FDR statements are true on average (*in expectation*) over all possible experiments that could have been performed, *post hoc* inference statements are true *with probability* $1-\alpha$. For example, if for $\alpha = 0.05$ we find $\gamma(C) = 0.875$ for a given cluster C, it means that we are 95% confident that more than 87.5% of the pixels in C are different between the two conditions.
- Second, standard FDR controlling procedures are not applicable to data-driven pixel clusters (Goeman and Solari 2011). In contrast, *post hoc* bounds provide statistical guarantees for any number of user-defined subset of hypotheses (here, pixel clusters).

Here, we use the Simes bound introduced by Goeman and Solari (2011) and defined as $\gamma_\alpha : C \mapsto 1 - V_\alpha(C)/|C|$, where

$$V_\alpha(C) = \min_{1 \leq n \leq |C|} \Big[\sum_{(i,j) \in C} \mathbb{1}_{\{\mathbf{P}_{ij} > \frac{\alpha n}{m}\}} + n - 1 \Big]. \tag{2}$$

The formulation of Equation (2) was obtained by Blanchard *et al.* (2020) and a linear time implementation is described in Enjalbert-Courrech and Neuvial (2022). It only requires the availability of a *P*-value $\mathbf{P}_{ij}$ corresponding to positively dependent

Downloaded from academic.oup.com/bioinformatics/article/42/9/btag631/8770358 by guest on 24 September 2026

(PRDS) test statistics as defined in Benjamini and Yekutieli (2001). As such, it is valid under the same assumptions as the widely used BH procedure for FDR control.

2.3 Results evaluation

2.3.1 Semi-synthetic data benchmark

We first evaluated *hicream* using semi-synthetic data with a controlled ground-truth signal, as done previously (Jorge *et al.* 2025b). Briefly, real Hi-C contact matrices from the same biological condition (for which no differences are expected) were used as a null hypothesis baseline. Artificial signals were then introduced into specific local regions of a subset of matrices to assess the capacity of the methods to detect them. More specifically, we used five technical replicates of a Hi-C library from a human colon sample (The ENCODE Project Consortium 2012). Four of the five technical replicates were used to form two artificial groups of duplicates, while the fifth replicate was used to add counts in predefined regions of the Hi-C matrices, creating known differences between the two groups. This process is illustrated in Fig. S2, available as supplementary data at *Bioinformatics* online.

We tested and compared the two versions of *hicream* and the approach proposed in *diffHic clustering*. Clusters of interest were then selected as:

- *hicream*: clusters for which the *post hoc* bound (for $\alpha = 0.05$) was larger than a given threshold (i.e. as in step (D) of Fig. 1);
- *diffHic clustering*: clusters for which the *cluster-level FDR* computed by the package was smaller than a given threshold.

These two threshold values are set independently of each other since the two criteria are not comparable.

For a given threshold and method, we obtained the precision as the proportion of pixels correctly declared differential among all pixels in selected clusters. The recall (power) was also computed similarly. Precision-Recall (PR) curves were obtained by varying the selection threshold from 5% to 100% for *diffHic clustering* and for each unique observed value of γ_α for *hicream*. A perfect classifier would yield a precision of 1 (no false positives) and a recall of 1 (no false negatives). Therefore, the "best" clustering for each method was obtained as the (Precision, Recall) point closest to $(1, 1)$ (which corresponds to two distinct thresholds for the two methods). The evaluation was carried out for three chromosomes (1, 7, and 21) at three resolutions (200 kb, 500 kb, and 1 Mb).

2.3.2 Application to real use case datasets

The first dataset comes from a Hi-C study conducted on a murine cell line under two condition: either CTCF-depleted (CTCF- condition) or control (CTCF+ condition) (Zhang *et al.* 2021). Additional data from ChIP-seq experiments against CTCF on the same cell line were also used (Zhang *et al.* 2019). Hi-C matrices and ChIP-seq narrow peaks were downloaded from GEO (accessions GSE168251 and GSE129997).

The second real dataset comes from a comparison of mouse cell lines at different stages of neuronal differentiation (Bonev *et al.* 2017). Raw Hi-C data from stem cells (ES) and cortical neurons (CN) were downloaded from GEO (accession GSE96107) and processed with the *nfcore/hic* pipeline to obtain four matrices (replicates) of each condition at 50 kb resolution. Corresponding ChIP-seq data were obtained from the UCSC Table Browser.

Each dataset was processed with *hicream diff*. To better characterize the shape of obtained clusters and profile their geometrical diversity with respect to their TDP, we designed three shape-related metrics and assigned them to each cluster: a triangle-, a stripe-, and a square-shape score. Relationships between shape metrics, TDP, and absolute log-fold change (logFC) were characterized using Principal Component Analysis (PCA).

Then, clusters with a minimal TDP greater than 10% across all chromosomes were selected for further analysis.

Clusters selected in the neuronal differentiation study were compared with the results obtained from two other tools designed to identify significant differences in TADs (*TADCompare*, Cresswell and Dozmorov [2020]) or compartments (*dcHiC*, Chakraborty *et al.* [2022]). Unlike *hicream*, both tools identify 1D location on the genome corresponding to changes. Hence, selected *hicream* clusters were projected onto the genome to identify the corresponding genomic boundaries and their flanking bins. Then, to assess the consistency between *hicream* and *TADCompare*, we tested for the enrichment of these cluster boundaries in differential shifted regions identified by *TADCompare*, compared to flanking bins. As for *dcHiC*, we explored the relationship, at the pixel level, between TDP (related to the cluster in which the pixel is included) and major changes in compartment conformation (when a pixel is between two "A" regions in condition 1 and between two "B" regions in condition 2, or the opposite).

hicream clusters of both studies were also compared with known biological signal related to genomic organization: We tested for CTCF enrichment at cluster boundaries compared with the flanking bins, by using data on active CTCF binding sites from ChIP-seq experiments. In addition, the distribution of active CTCF binding sites was further characterized by profiling their relative positions with respect to the cluster boundaries.

Finally, cluster boundaries were also tested for enrichment of all known TFBSs from the JASPAR database (CORE Vertebrates 2024) using the AME tool from the MEME Suite (v5.5.4). Last, genes with enriched binding motifs from the TFBS analysis were also subjected to a Gene Ontology functional analysis using the Panther Classification System tool from https://geneontology.org (PANTHER GO-Slim Biological Process, Fisher exact test and Bonferroni multiple testing correction).

Further information on data processing and cluster validation is provided in Supplementary Section S2, available as supplementary data at *Bioinformatics* online.

2.4 Package implementation and visualization

hicream is implemented as an R package available on CRAN (Jorge *et al.* 2025a). Since the default plotting functions provide only static visualizations, we additionally developed an interactive interface based on the R package *animint2* (Sievert *et al.* 2019). All results from this study can be explored online at https://eoojj.github.io/hicream-article-viz/. A detailed illustration of this interactive visualization platform is provided in Fig. S3, available as supplementary data at *Bioinformatics* online.

Downloaded from academic.oup.com/bioinformatics/article/42/9/btag631/8770358 by guest on 24 September 2026

3 Results

3.1 On semi-synthetic data, hicream identifies relevant clusters of differential pixels

Figure 2 displays Precision-Recall curves for the three compared approaches (both *hicream* versions and *diffHic clustering*) on the semi-synthetic dataset. *hicream diff* systematically outperforms *diffHiC clustering* and *hicream counts*. *hicream counts* outperforms *diffHic clustering* at 200 kb resolution but does not show better performance at 500 kb or 1 Mb resolution. As the *hicream counts* variant does not outperform the default *hicream diff* method, we chose to focus on *hicream diff* only in the sequel.

Table 1 gives some characteristics of the "best" selection for each chromosome and resolution for *hicream diff* and *diffHic clustering*. Across all resolutions and chromosomes, *hicream diff* can reach high performance (in terms of precision and recall) with a small number of clusters compared to *diffHic clustering*. Indeed, as can be observed for chromosome 1 at 200 kb, *hicream diff* extracts only 120 clusters—118 times less than *diffHic clustering*—with a median size of 36 pixels, which is 12 times larger than *diffHic clustering*. Moreover, for chromosome 21 at 1 Mb, despite showing similar performance, both methods largely differ in their output clustering as *hicream diff* summarizes the results in only 7 clusters, while *diffHic clustering* yields 100 clusters, most of them being composed of only one single pixel. Altogether, these results highlight the agreggating and smoothing properties of *hicream*, which allows to properly cluster pixels, facilitating result interpretation.

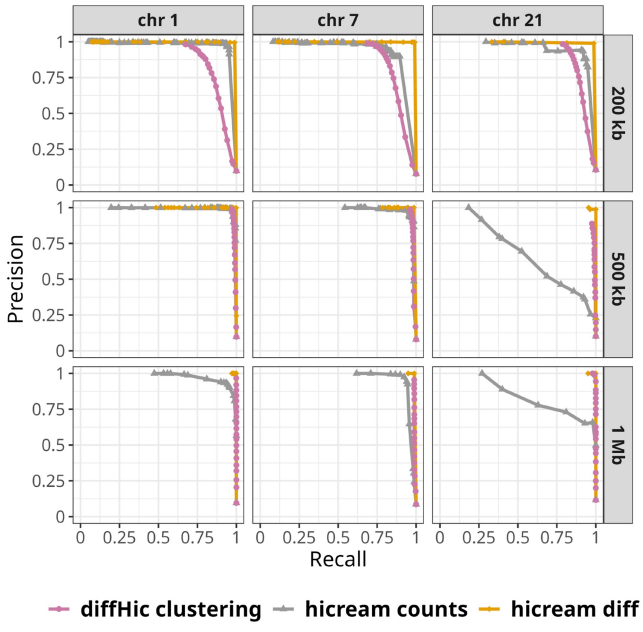


Figure 2 PR curves (semi-synthetic data) computed for three chromosomes of contrasted lengths (chr 1, 7, and 21), at three different bin resolutions (200 kb, 500 kb, and 1 Mb), and for the three methods (both *hicream* versions and *diffHic clustering*). Every point corresponds to a different given threshold value (see Section 2.3.1).

3.2 On real data, hicream identifies clear and contrasted clusters of various shapes

By applying *hicream* to real Hi-C datasets, we found that our method reveals sharp clusters from pixel-based results of differential analysis. Figure 3 illustrates these observations in the CTCF depletion data: the upper half displays the cluster-based minimal TDP obtained by *hicream diff*, while the lower half shows the original pixel-based adjusted *P*-values derived from *diffHic* (chromosome 13, 200 kb). High-TDP clusters in the upper half typically align with groups of pixels exhibiting significant adjusted *P*-values in the lower half, with sharper and more distinct boundaries. This contrast is improved by the fact that significant yet isolated pixels in the lower half usually get assigned to large clusters with low TDP bounds in the upper half. This illustrates how *hicream* can extend the results of *diffHic*, by removing the background noise, highlighting the most differential regions and providing statistical guarantees.

Importantly, the resulting differential clusters exhibit a wide range of shapes and sizes, showing greater diversity than is typically observed in Hi-C interaction matrices (Figs 3 and 4). Whereas Hi-C data are generally characterized by a hierarchical TAD-like organization, the focused and precise delineation of groups of differentially proximal interactions makes it possible to highlight the variable components of these canonical structures. In this context, the ability of *hicream* to detect statistically significant clusters of arbitrary shape without imposing *a priori* constraints constitutes a valuable feature.

Figure S4, available as supplementary data at *Bioinformatics* online further illustrates the diversity of shapes identified by our method with a PCA of shape metrics, TDP and logFC. It exhibits substantially different patterns across biological datasets. Interestingly, on the CTCF depletion study, high TDPs are associated with large stripe-scores, consistent with the chromatin extrusion process underlying TAD boundary and chromatin loop formation. On the contrary, in the neuronal differentiation study, high TDPs are associated with large triangle-scores, showing that a different biological process (TAD compaction or de-compaction) is at hand in this dataset. Overall, this supports the notion that various patterns of 3D variation drive distinct functional effects, and suggests that *hicream* can capture a broad range of dynamic conformations that are not limited to a specific structure.

To confirm this generality of application, we compared *hicream* clusters with predictions from two structure-level differential methods, both run on the neuronal differentiation data: *TADCompare* to detect variable TAD boundaries, and *dcHiC* to detect differences in chromatin compartmentalization.

Figure S5, available as supplementary data at *Bioinformatics* online shows the enrichment of bins identified as differentially shifted by *TADCompare* and enriched in CN (resp. ES) condition at positive (resp. negative) log-fold change cluster boundaries. As a negative control, the same figure shows a depletion of differentially shifted bins enriched in CN (resp. ES) at negative (resp. positive) log-fold change cluster boundaries.

To compare with compartment differences, we extracted pixels found significantly involved in a major compartment change by *dcHiC*: These were pixels "AA2BB" (between two bins in "A" compartment for the ES condition and in "B" compartment for

Downloaded from academic.oup.com/bioinformatics/article/42/9/btag631/8770358 by guest on 24 September 2026

Table 1 Characteristics of best performing clusterings (semi-synthetic data).

		diffHic clustering					hicream diff				
Chr.	Res.	Cluster-level FDR	Nb. of clusters	Median cluster size	Precision	Recall	$\gamma_\alpha(\cdot)$ threshold	Nb. of clusters	Median cluster size	Precision	Recall
1	200 kb	0.40	14198	3	0.879	0.802	0.014	120	36.0	0.996	0.989
1	500 kb	0.05	3037	4	0.988	0.971	0.086	72	70.0	0.998	0.999
1	1 Mb	0.05	2634	1	0.967	0.999	0.200	76	10.5	1.000	1.000
7	200 kb	0.45	3994	4	0.904	0.802	0.005	76	33.0	0.997	0.991
7	500 kb	0.05	1131	4	0.987	0.962	0.258	153	10.0	1.000	0.986
7	1 Mb	0.05	1109	1	0.955	0.987	0.698	42	11.0	1.000	0.988
21	200 kb	0.25	314	5	0.939	0.833	0.291	21	34.0	0.989	0.988
21	500 kb	0.05	151	2	0.888	0.975	0.083	19	10.0	0.988	1.000
21	1 Mb	0.10	100	1	0.990	0.990	0.200	7	12.0	1.000	1.000

The best performance corresponds to the (precision, recall) point closest to $(1,1)$. For all chromosomes and resolutions and for both *diffHic clustering* and *hicream diff*, information about clustering statistics (number of clusters and their median size in number of pixels) as well as corresponding precision and recall are provided. For *hicream diff*, *post hoc* inference was applied with $\alpha = 0.05$ and the threshold used for cluster selection is given in the column "$\gamma_\alpha(\cdot)$ threshold." Two particular cases are circled in blue and commented in the text.

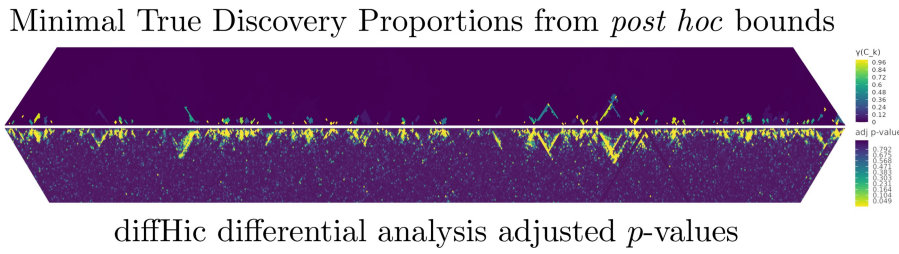
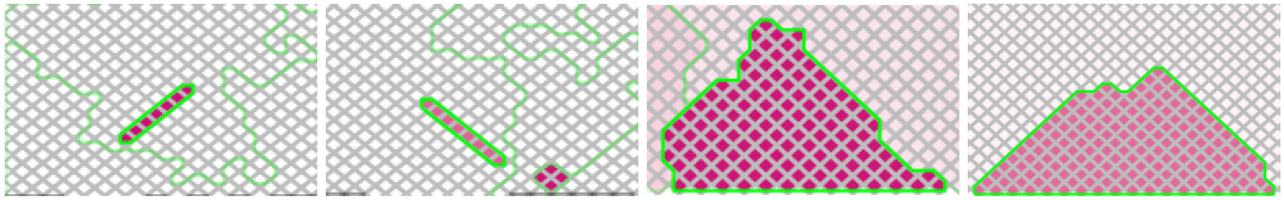


Figure 3 Illustration of hicream smoothing and aggregating properties (CTCF depletion study). *hicream diff* was applied to chromosome 13 of the CTCF depletion dataset at 200 kb resolution. The *kneebow* rule selected 1261 clusters. Upper half: *hicream diff* results. Each cluster of pixels C_k is displayed with a color corresponding to its minimal proportion of true positives $\gamma(C_k)$. Lower half: adjusted *P*-values obtained from *diffHic* differential analysis. The Hi-C matrix representation has been truncated to focus on the diagonal.



(a) CTCF depletion study
Chr 1 - Cluster 468
31.2 Mb to 34 Mb
$|C_{468}| = 7$, $\gamma(C_{468}) = 1$
$|\texttt{stripe_score}| = 1.946$
$q_{C_{468}}^{|\texttt{stripe_score}|} = 0.999$

(b) CTCF depletion study
Chr 14 - Cluster 310
112.8 Mb to 115.2 Mb
$|C_{310}| = 8$, $\gamma(C_{310}) = 0.75$
$|\texttt{stripe_score}| = 2.079$
$q_{C_{310}}^{|\texttt{stripe_score}|} = 0.995$

(c) ND study
Chr 3 - Cluster 60
89.35 Mb to 90.25 Mb
$|C_{60}| = 124$, $\gamma(C_{60}) = 0.99$
$\texttt{triangle_score} = 0.765$
$q_{C_{60}}^{\texttt{triangle_score}} = 0.998$

(d) ND study
Chr 6 - Cluster 373
68 Mb to 68.95 Mb
$|C_{373}| = 173$, $\gamma(C_{373}) = 0.72$
$\texttt{triangle_score} = 0.958$
$q_{C_{373}}^{\texttt{triangle_score}} = 0.997$

Figure 4 Stripe and triangle -shaped differential clusters identified by hicream (CTCF depletion and Neuronal Differentiation studies). The green lines are cluster boundaries and each cluster C_k is colored according to $\gamma_\alpha(C_k)$. Four selected clusters, highlighted with a slightly more vivid boundary color, are examples of stripe-shaped differential clusters identified with *hicream diff* in the CTCF depletion study at 200 kb resolution, and triangle-shaped differential clusters identified in the neuronal differentiation dataset at 50 kb resolution. For each cluster, information about chromosome, cluster number, cluster position on the chromosome (the left and right positions of the first and last chromosome bins, respectively, among the pixels belonging to this cluster), size (number of pixels), minimal True Discovery Proportion $\gamma_\alpha(C_k)$, shape score (either stripe_score or triangle_score), and quantile of the shape score for the corresponding cluster is given below the plot.

the CN condition) and "BB2AA" (that correspond to the opposite situation). Figure S6, available as supplementary data at *Bioinformatics* online shows the distribution of cluster TDP for these specific pixels compared to all other pixels and exhibits a significantly increased TDP for pixels corresponding to compartment changes (Wilcoxon test, *P*-values $< 2.22e-16$).

The link between compartment differences and differential clusters identified by *hicream* is not limited to "AA2BB" changes

Downloaded from academic.oup.com/bioinformatics/article/42/9/btag631/8770358 by guest on 24 September 2026

only. Indeed, Fig. S7, available as supplementary data at *Bioinformatics* online displays a group of 7 differential clusters (TDP $> 10\%$) identified by *hicream* on chromosome 19 of the neuronal differentiation study that represent an "AA2AB" switch. The left projections of the 7 clusters are all located in a region of non-differential bins corresponding to the A compartment in both ES and CN conditions. On the other hand, the right projections of the clusters contain 3 differential bins that switch from A compartment in ES condition to B compartment in CN condition. Therefore, those clusters correspond to pixels representing AA interactions in ES condition and AB interactions in CN condition. This is consistent with the fact that the 7 clusters have a negative log-fold change.

These results confirm that although *hicream* is neither designed nor limited to find differences in TADs or compartments specifically, it is able to identify signal related to multiple levels of chromatin organization at once, showing consistency with structure-level prediction tools.

3.3 Differential clusters identified by hicream show relevant TFBS enrichment at their boundaries

We further investigated the biological relevance of *hicream* clusters by focusing on the boundaries of their genomic projections in both datasets. First, using experimental ChIP-seq data, we observed a significant enrichment of active CTCF binding sites at cluster boundaries compared to their flanking regions, with an average of 21.1 peaks versus 16.8 in the neuronal differentiation study, and 5.7 peaks versus 3.8 in the CTCF depletion study (P-value $= 2.52 \times 10^{-16}$ and 6.26×10^{-8} respectively, Wilcoxon paired test; see also Fig. S8, available as supplementary data at *Bioinformatics* online for the neuronal differentiation study).

For the CTCF depletion study, we further examined the full CTCF occupancy profiles within and around the projected clusters depending on the sign of their logFC. As shown in Figure 5, the occupancy profiles differ strikingly between clusters with positive and negative logFC values, exhibiting complementary patterns. This difference is consistent with the known effects of CTCF depletion on topologically associating domains (TADs) (see Fig. S9 and Supplementary Section S3, available as supplementary data at *Bioinformatics* online). In brief, CTCF depletion is known to weaken TAD boundary insulation, allowing increased inter-TAD interactions across boundaries. As a result, pixels with increased interaction values after depletion are expected to localize above and around annotated TAD boundaries in Hi-C matrices, giving rise to clusters with positive logFC values and high CTCF binding site densities. Conversely, as CTCF depletion globally reduces TAD strength, intra-TAD interactions are expected to decrease. This effect generates differential clusters with negative logFC values and TAD-like CTCF profiles at their boundaries.

Last, we extended the enrichment analysis to all known transcription factor binding sites (TFBSs) from the JASPAR database (see Methods). For the neuronal differentiation study, CTCF showed the most significant enrichment (Supplementary File S2, available as supplementary data at *Bioinformatics* online). Functional analysis of all enriched transcription factors revealed an overrepresentation of Gene Ontology terms related to cell differentiation and neuronal processes, including regulation of neurogenesis, regulation of cell differentiation, brain development, axon development, and other neuronal functions (Table S1, available as supplementary data at *Bioinformatics* online).

For the CTCF depletion study, in contrast to the results obtained with the ChIP-seq data, no significantly enriched TFBS were identified. Since ChIP-seq data reflect active CTCF binding sites, whereas TFBS enrichment is based on computational predictions, this discrepancy may suggest that *hicream* more accurately identifies sites that are truly affected by the depletion experiment. Together, these results support the biological relevance of *hicream* differential clusters and demonstrate that *hicream* accurately captures spatial reorganization of chromatin interactions from real Hi-C data.

4 Discussion and conclusion

We have introduced *hicream*, a flexible three-step framework of Hi-C data differential analysis. On a semi-synthetic dataset with controlled ground truth, we compared both versions of *hicream* with the only other method that enables data-driven discovery of differential clusters in Hi-C data, *diffHic* clustering. Across all tested chromosomes and resolutions, both versions of *hicream* outperformed *diffHic clustering*. Fewer but larger differential clusters were obtained, highlighting its ability to capture coherent and interpretable differential genomic structures.

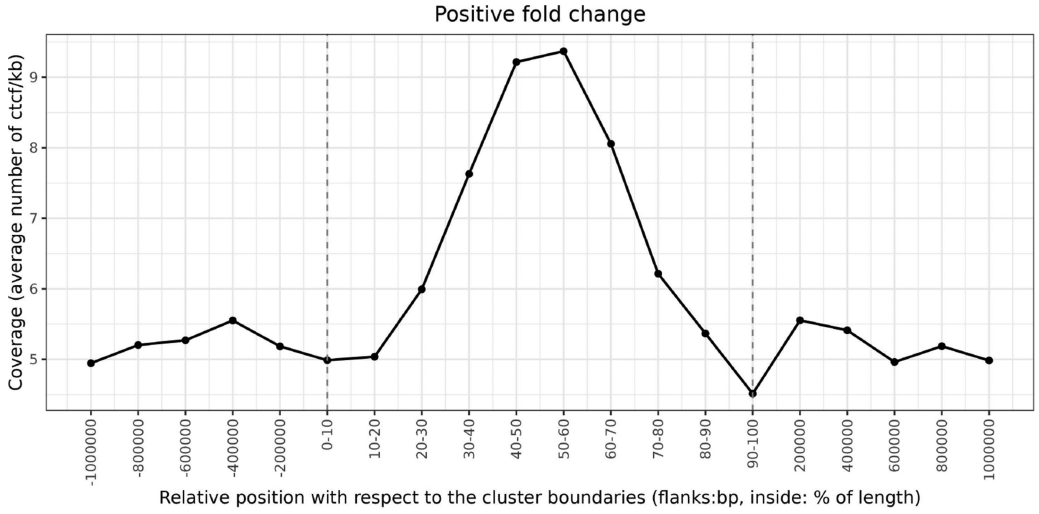
In real case studies, we further showed that *hicream* can detect differential clusters of arbitrary shape, including non-standard yet biologically relevant structures such as stripes. Moreover, the results were consistent with the distribution of active CTCF binding sites and TFBSs at cluster boundaries, further supporting the ability of *hicream* to accurately identify reorganized genomic regions.

Furthermore, the method is highly generic and modular: The method (and implementation) can accommodate alternative choices for differential analysis and clustering than those implemented in *hicream*. In particular, the differential analysis step can be extended to more complex experimental designs by incorporating covariates and testing specific hypotheses through flexible contrasts.

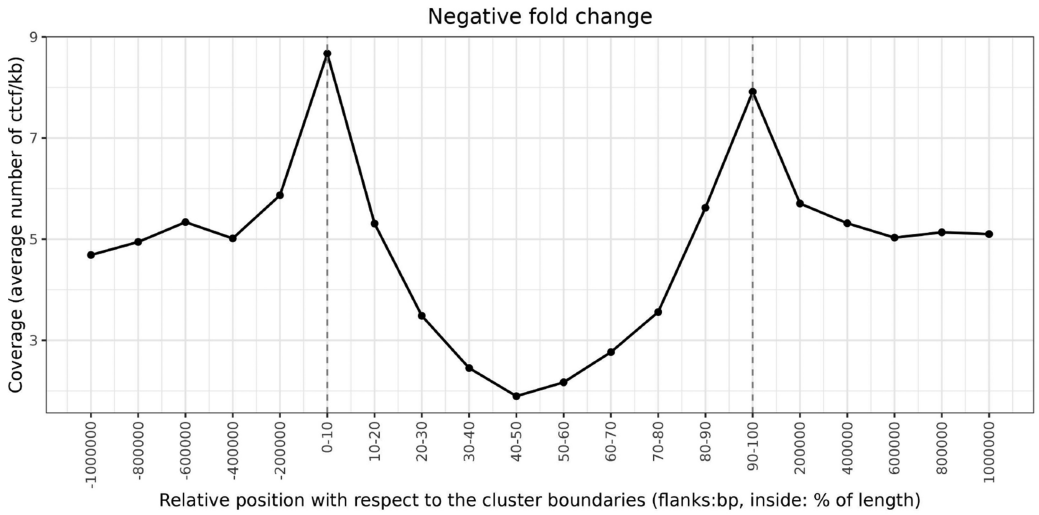
Moreover, because *post hoc* inference provides statistical guarantees independently of the clustering procedure, prior knowledge can also be used to define regions of interest. For example, candidate regions may be derived from structural predictions such as TADs, A/B compartments, or chromatin loops when targeting specific genomic structures. Prior knowledge could also be integrated into the clustering procedure itself, by constraining it to produce clusters of a specific shape (e.g. stripe-like). While the former is straightforward to implement in *hicream* as it only requires specifying candidate regions, the latter would entail designing new clustering procedures able to enforce structural constraints, which constitutes a stimulating methodological perspective for the present work.

Finally, the *post hoc* inference step currently relies on the Simes bound, which may be conservative. In settings with spatial dependence, a calibration strategy has been described to

Downloaded from academic.oup.com/bioinformatics/article/42/9/btag631/8770358 by guest on 24 September 2026



(a) Positive log-fold change clusters



(b) Negative log-fold change clusters

Figure 5 Distribution of CTCF peak coverage within and around the projected regions of hicream differential clusters at 200 kb resolution (CTCF depletion study). Positions on the *x*-axis correspond to left and right flanking regions of the projected boundaries and ten genomic intervals of size 10% of each cluster length between the two boundaries. Top (resp. Bottom): Clusters with a majority of positive (resp. negative) log-fold change pixels.

obtain sharper bounds (Blanchard *et al*. 2020), which is a challenging but interesting direction for future research.

Acknowledgments

We would like to thank the Genotoul-Bioinfo platform (Toulouse Occitanie) for providing computational resources and assistance.

Author contributions

Elise Jorge (Conceptualization [Equal], Data curation [Equal], Formal analysis [Lead], Funding acquisition [Equal], Investigation [Equal], Methodology [Lead], Project administration [Equal], Software [Lead], Validation [Equal], Visualization [Equal], Writing—original draft [Lead], Writing—review & editing [Equal]), Toby Dylan Hocking (Funding acquisition [Supporting], Resources [Supporting], Software [Supporting], Supervision [Supporting], Visualization [Lead], Writing—review & editing [Supporting]), Pierre Neuvial (Conceptualization [Equal], Data curation [Equal], Formal analysis [Equal], Funding acquisition [Equal], Investigation [Equal], Methodology [Equal], Project administration [Lead], Resources [Lead], Software [Supporting], Supervision [Lead], Validation [Equal], Visualization [Equal], Writing—original draft [Supporting], Writing—review & editing [Equal]), Nathalie Vialaneix (Conceptualization [Equal], Data curation [Equal], Formal analysis [Equal], Funding acquisition [Lead], Investigation [Equal], Methodology [Equal], Project administration [Lead], Resources [Lead], Software [Equal], Supervision [Lead], Validation [Equal], Visualization [Equal], Writing—original draft [Equal],

Downloaded from academic.oup.com/bioinformatics/article/42/9/btag631/8770358 by guest on 24 September 2026

Writing—review & editing [Equal]), and Sylvain Foissac (Conceptualization [Equal], Data curation [Equal], Formal analysis [Equal], Funding acquisition [Lead], Investigation [Equal], Methodology [Equal], Project administration [Lead], Resources [Lead], Software [Supporting], Supervision [Lead], Validation [Equal], Visualization [Equal], Writing—original draft [Supporting], Writing—review & editing [Equal])

Conflicts of interest

None declared.

Funding

This work was supported by the INRAE DIGIT-BIO program and by a Toulouse INP-ENSAT fellowship.

Data availability

All datasets used in this article are available or described at https://doi.org/10.57745/IEUSLV. The scripts reproducing the results of this manuscript are all provided at https://forge.inrae.fr/scales/replication-repository-for-hicream-article.

References

Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J R Stat Soc Ser B (Methodological)* 1995;**57**:289–300.

Benjamini Y, Yekutieli D. The control of the false discovery rate in multiple testing under dependency. *Ann Statist* 2001;**29**:1165–88. https://doi.org/10.1214/aos/1013699998

Blanchard G, Neuvial P, Roquain E. Post hoc confidence bounds on false positives using reference families. *Ann Statist* 2020;**48**:1281–303. https://doi.org/10.1214/19-AOS1847

Bonev B, Cohen NM, Szabo Q *et al.* Multiscale 3D genome rewiring during mouse neural development. *Cell* 2017;**171**:557–72.e24. https://doi.org/10.1016/j.cell.2017.09.043

Chakraborty A, Wang JG, Ay F. dcHiC detects differential compartments across multiple Hi-C datasets. *Nat Commun* 2022;**13**:6827. https://doi.org/10.1038/s41467-022-34626-6

Cresswell KG, Dozmorov MG. TADCompare: an R package for differential and temporal analysis of topologically associated domains. *Front Genet* 2020;**11**:158. https://doi.org/10.3389/fgene.2020.00158

Enjalbert-Courrech N, Neuvial P. Powerful and interpretable control of false discoveries in differential expression studies. *Bioinformatics* 2022;**38**:5214–21. https://doi.org/10.1093/bioinformatics/btac693

Gjoni K, Gunsalus LM, Kuang S *et al.* Comparing chromatin contact maps at scale: methods and insights. *Nat Methods* 2025;**22**:824–33. https://doi.org/10.1038/s41592-025-02630-5

Goeman JJ, Solari A. Multiple testing for exploratory research. *Statist Sci* 2011;**26**:584–97. https://doi.org/10.1214/11-STS356

Hua D, Gu M, Zhang X *et al.* DiffDomain enables identification of structurally reorganized topologically associating domains. *Nat Commun* 2024;**15**:502. https://doi.org/10.1038/s41467-024-44782-6

Jorge E, Foissac S, Hocking T *et al. hicream: HIC diffeREntial Analysis Method.* 2025a. http://CRAN.R-project.org/package=hicream. R package version 0.0.2 (2025-11-19), published on CRAN. Maintainer. Source code at https://forge.inrae.fr/scales/hicream.

Jorge E, Foissac S, Neuvial P *et al.* A comprehensive review and benchmark of differential analysis tools for Hi-C data. *Brief Bioinform* 2025b;**26**:bbaf074. https://doi.org/10.1093/bib/bbaf074

Lareau CA, Aryee MJ. diffloop: a computational framework for identifying and analyzing differential DNA loops from sequencing data. *Bioinformatics* 2018;**34**:672–4. https://doi.org/10.1093/bioinformatics/btx623

Lieberman-Aiden E, Van Berkum NL, Williams L *et al.* Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* 2009;**326**:289–93. https://doi.org/10.1126/science.1181369

Lun AT, Smyth GK. diffHic: A bioconductor package to detect differential genomic interactions in Hi-C data. *BMC Bioinformatics* 2015;**16**:258. https://doi.org/10.1186/s12859-015-0683-0

Lupiáñez DG, Kraft K, Heinrich V *et al.* Disruptions of topological chromatin domains cause pathogenic rewiring of gene-enhancer interactions. *Cell* 2015;**161**:1012–25. https://doi.org/10.1016/j.cell.2015.04.004

Maigné E, Kurylo C, Zytnicki M *et al.* HiCDOC: chromatin compartment prediction and differential analysis from Hi-C data with replicates. bioRxiv, https://doi.org/10.1101/2025.09.18.677058, 2025, preprint: not peer reviewed.

Mourad R. TADreg: a versatile regression framework for TAD identification, differential analysis and rearranged 3D genome prediction. *BMC Bioinformatics* 2022;**23**:82. https://doi.org/10.1186/s12859-022-04614-0

Oudelaar AM, Beagrie RA, Gosden M *et al.* Dynamics of the 4D genome during in vivo lineage specification and differentiation. *Nat Commun* 2020;**11**:2722. https://doi.org/10.1038/s41467-020-16598-7

Pedregosa F, Varoquaux G, Gramfort A *et al.* Scikit-learn: machine learning in python. *J Machine Learn Res* 2011;**12**:2825–30.

Randriamihamison N, Vialaneix N, Neuvial P. Applicability and interpretability of Ward's hierarchical agglomerative clustering with or without contiguity constraints. *J Classif* 2021;**38**:363–89. https://doi.org/10.1007/s00357-020-09377-y

Sievert C, VanderPlas S, Cai J *et al.* Extending ggplot2 for Linked and Animated Web Graphics. *J Comput Graph Stat* 2019;**28**:299–308. https://doi.org/10.1080/10618600.2018.1513367

Spielmann M, Lupiáñez DG, Mundlos S. Structural variation in the 3D genome. *Nat Rev Genet* 2018;**19**:453–67. https://doi.org/10.1038/s41576-018-0007-0

The ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature* 2012;**489**:57–74. https://doi.org/10.1038/nature11247

Zhang H, Emerson DJ, Gilgenast TG *et al.* Chromatin structure dynamics during the mitosis-to-G1 phase transition. *Nature* 2019;**576**:158–62. https://doi.org/10.1038/s41586-019-1778-y

Zhang H, Lam J, Zhang D *et al.* CTCF and transcription influence chromatin structure re-configuration after mitosis. *Nat Commun* 2021;**12**:5157. https://doi.org/10.1038/s41467-021-25418-5

Downloaded from academic.oup.com/bioinformatics/article/42/9/btag631/8770358 by guest on 24 September 2026

© The Author(s) 2026. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.
Bioinformatics, 2026, 42, 1–9
https://doi.org/10.1093/bioinformatics/btag631
Original Paper

Downloaded from academic.oup.com/bioinformatics/article/42/9/btag631/8770358 by guest on 24 September 2026

Chapitre 4

Conclusions et perspectives

Le travail réalisé dans cette thèse est une contribution méthodologique et statistique à l'identification de réarrangements de structures tridimensionnelles dans le génome. En particulier, en s'appuyant sur des données fournies par l'expérience Hi-C, ce travail a permis d'évaluer des outils existants d'analyse différentielle basée sur les pixels ainsi que de proposer une nouvelle approche d'analyse différentielle, nommée **hicream**.

Méthodologie d'évaluation des outils d'analyse différentielle. Dans une première partie, motivée par le manque d'évaluation et de comparaison objective des méthodes d'analyse différentielle de données Hi-C basées sur les pixels, nous avons proposé un benchmark des méthodes existantes. Quand un nouvel outil de ce type est publié, le papier présente généralement une ou des comparaisons avec des outils existants. Or, ce type de comparaison, limité à chaque fois à quelques outils, dans un cadre bien particulier (choix des données, métriques utilisées pour l'évaluation, etc.) ne permet pas de tirer de conclusions générales sur la qualité et l'efficacité des méthodes. Ces remarques sont également valables pour les autres outils d'analyse différentielle de données Hi-C, en particulier ceux basés sur l'identification de structures différentielles. Ainsi, au fur et à mesure que de nouvelles méthodes apparaissent, il est d'intérêt pour les utilisatrices d'avoir accès à des travaux présentant la comparaison et l'évaluation rigoureuse des méthodes existantes afin de les guider dans le choix d'un ou plusieurs outils à utiliser.

En ce qui concerne les méthodes d'analyse différentielle de données Hi-C, une des manières les plus couramment utilisées de valider les résultats est de comparer les détections positives à des signaux biologiques. Comme mentionné dans l'introduction, certains éléments – tels que la protéine CTCF ou la cohésine – sont connus pour être associés à la formation de structures. La distribution des détections positives (pixels et/ou régions différentielles) relativement à la présence de ces éléments le long du génome est alors évaluée au travers de données ChIP-Seq par exemple. Cependant, bien que ce type de résultat puisse illustrer la pertinence biologique des détections, cela ne permet pas d'évaluer la qualité d'un modèle et du test associé d'un point de vue statistique.

Afin de pouvoir réaliser une évaluation rigoureuse d'outils d'analyse différentielle, il faudrait disposer de matrices Hi-C correspondant à deux conditions différentes et sur

lesquelles la position des pixels différentiels est connue. Par ailleurs, comme plusieurs méthodes d'analyse différentielle basées sur les pixels consistent à comparer deux groupes de plusieurs matrices Hi-C contenant des réplicats biologiques, il est d'intérêt d'avoir à sa disposition des données qui représentent fidèlement la variabilité intra-condition que l'on retrouve dans les données Hi-C.

Les techniques existantes de simulation de données Hi-C, dont l'objectif est souvent de réduire les coûts associés aux traitements et manipulations, peuvent consister à produire des matrices Hi-C simplement à partir de la séquence d'ADN ou de données biologiques relatives à la séquence (pics CTCF, marques d'histones, etc.) [Zhang et al., 2019, Fudenberg et al., 2020, Schwessinger et al., 2020, Yang et al., 2023]. Ces méthodes, qui se basent généralement sur des données d'apprentissage pour prédire des matrices Hi-C, peuvent permettre d'obtenir des données à des résolutions fines, qui sont les plus coûteuses. Cependant, avec ces outils, il ne semble actuellement pas possible de simuler de la variabilité intra-condition dans les matrices Hi-C.

La méthode **FreeHi-C** [Zheng and Keleş, 2020] consiste, elle, à utiliser un fichier brut issu du séquençage d'une librairie Hi-C et à apprendre, en appliquant les étapes d'alignement, d'association des paires de lectures, etc., des métriques de description de la librairie qui peuvent ensuite être utilisées pour produire des données simulées. Au moment de la rédaction de ce manuscrit, j'ai pris connaissance de l'existence d'une extension de cette méthode de simulation [Zheng et al., 2021] qui propose de créer des données sous H_1 avec des signaux différentiels ajoutés ponctuellement dans les matrices. De plus, ce papier propose d'utiliser cette approche de simulation pour la comparaison de quatre outils (également évalués dans notre benchmark). Il est rassurant de constater que notre benchmark ainsi que ce dernier papier retrouvent des résultats concordants sous H_0 en indiquant que seules les méthodes **diffHiC** [Lun and Smyth, 2015] et **multiHiCcompare** [Stansfield et al., 2019] contrôlent l'erreur de type I. Par ailleurs, l'outil **Clipper** [Ge et al., 2021], qui propose de détecter des pixels différentiels sans calcul de p-valeurs à travers l'utilisation de permutations, a également réalisé une comparaison d'outils d'analyse différentielle basée sur les pixels. En termes de contrôle du FDR, **Clipper** indique un bon contrôle par **diffHiC** tandis que cette affirmation est plus nuancée pour le benchmark **FreeHi-C** et le nôtre, qui indiquent un défaut de contrôle global avec néanmoins un contrôle satisfaisant par **diffHiC** sur certaines configurations de tests et certains niveaux de FDR ciblés. Enfin, en ce qui concerne la puissance de détection, les performances de chaque outil s'avèrent plutôt différentes entre les trois papiers. Par exemple, la méthode **Selfish** [Ardakany et al., 2019], qui a démontré très peu de puissance de détection sur nos tests, s'est avérée être la plus puissante des quatre testées dans le benchmark **FreeHi-C**. Cependant, il n'est pas incohérent d'observer une telle variabilité car les données H_1 ont été simulées de manière différente entre ces trois études.

Par ailleurs, comme nous l'avons déjà précisé dans le chapitre 2, si un outil ne contrôle pas l'erreur de type I et même si l'erreur de type II est faible, celui-ci n'est pas statistiquement valide et ne devrait donc pas être utilisé.

D'autres méthodes proposent de simuler des matrices Hi-C en modélisant spécifiquement certaines structures, c'est le cas par exemple de l'article évaluant la méthode **diffHic** [Lun and Smyth, 2015] qui propose de simuler des données avec des TADs différentiels. Il est proposé d'obtenir des données simulées selon une loi Négative Binomiale et c'est cette

même loi qui est utilisée par la méthode pour modéliser les comptages et réaliser le test de comparaison entre deux conditions. Il est donc important de noter qu'il peut exister un biais important sur les résultats en fonction de la similarité entre le modèle utilisé pour simuler les matrices et celui utilisé ensuite pour l'analyse différentielle. De manière plus générale, la difficulté de sélectionner un (ou plusieurs) jeu de données dans le cadre d'un benchmark est discutée dans Weber et al. [2019]. En particulier, sont abordées les problématiques propres aux jeux de données simulés comme leur manque de complexité ainsi que celles des jeux de données réels comme l'absence de « vérité terrain ». Par ailleurs, la nécessité de réaliser des évaluations sur un large spectre de jeux de données et avec diverses métriques afin de tirer des conclusions générales sur les performances des outils est mise en avant.

L'évaluation que nous avons proposée s'est basée sur des données « semi-simulées », c'est-à-dire sur des réplicats techniques de matrices Hi-C, à partir desquels nous avons pu créer un cas H_0 (pas de différences attendues entre les deux conditions) et un cas H_1 (différences présentes dans les données, à des positions connues). Dans le cas H_1, les pixels différentiels forment un triangle sur la diagonale de la matrice. Étant donné que nous n'avons testé que cette configuration spécifique pour H_1, les conclusions de cette étude ne peuvent être généralisées à tous les scénarios de réarrangements génomiques et donc à l'ensemble des structures différentielles possibles. En particulier, on peut s'attendre à ce que certaines méthodes soient plus ou moins capables de capturer des pixels différentiels en fonction de la distance à la diagonale, de la présence de pixels différentiels alentours, de la magnitude de la différence entre les deux conditions, etc.

Résultats clefs des comparaisons. Concernant les résultats obtenus sous H_0 – afin d'évaluer le contrôle de l'erreur de type I – il convient de noter qu'il existe un problème dans les sorties de certaines méthodes. En effet, certains outils ne fournissent pas les p-valeurs brutes mais uniquement les p-valeurs ajustées. De ce fait, il s'est avéré impossible de pouvoir évaluer la validité statistique de ces outils. Par ailleurs, nous avons noté que les corrections utilisées n'étaient pas toujours correctes, en proposant par exemple un contrôle du FDR relatif à la distance et donc n'étant pas assuré à l'échelle du chromosome. Il apparaît donc encore plus important de pouvoir fournir à l'utilisatrice les p-valeurs brutes, afin que celle-ci puisse leur appliquer le type de correction souhaité.

Les résultats du benchmark ont révélé une forte hétérogénéité de performance pour des méthodes utilisant le même type de modélisation statistique mais différant suivant les pré-traitements qu'elles utilisent. Ces pré-traitements, qui visent à filtrer des pixels et à corriger des biais dans les données via la normalisation, se révèlent donc primordiaux pour la qualité de l'analyse et des résultats. Nous avons même pu noter que, dans certains cas, certaines méthodes pouvaient filtrer tous les pixels (y compris ceux qui sont différentiels), ne permettant donc pas de réaliser l'analyse. Une perspective, probablement fort coûteuse d'un point de vue computationnel, serait de recenser tous les types de pré-traitements et tous les types de modélisations et tests utilisés afin d'évaluer toutes les combinaisons possibles. Cela permettrait ainsi de comprendre plus précisément le lien entre pré-traitements et tests statistiques ainsi que leur impact sur la qualité des résultats.

Ainsi, toutes ces remarques visent à souligner l'intérêt majeur des travaux de compa-

raison d'outils et soulèvent la problématique des données utilisées pour l'évaluation, sur l'exemple particulier de l'analyse différentielle de données Hi-C. À l'avenir, il serait souhaitable de pouvoir réaliser des comparaisons plus larges de ces outils en possédant des données Hi-C simulées incluant des régions différentielles de formes ou de types différents, potentiellement distantes de la diagonale, afin d'être plus exhaustives dans l'évaluation des performances statistiques de ces outils.

Intérêt et nouveauté de la méthode hicream. La méthode **hicream** introduit un cadre méthodologique statistiquement rigoureux pour réaliser l'analyse différentielle de données Hi-C. Par rapport aux approches existantes, elle offre la possibilité de pouvoir réaliser une analyse différentielle exploratoire et permet d'identifier, sur un même jeu de données, des clusters différentiels qui correspondent à des modifications de structures de types différents. En ce sens, elle peut, en principe, être utilisée quelle que soit la résolution des matrices en entrée. Cependant, il est à noter que des données à hautes résolutions – bien qu'elles permettent d'observer de fines structures telles que des boucles de chromatine – peuvent souffrir d'une faible couverture des bins et que la puissance statistique (de **diffHiC** par exemple) peut alors être limitée. Pour ces situations, on pourrait imaginer utiliser une procédure de simulation comme celle proposée dans **FreeHi-C** [Zheng and Keleş, 2020] afin d'augmenter artificiellement la profondeur de séquençage dans les matrices. Néanmoins, il faudrait pouvoir évaluer l'impact et les éventuels biais de ce type d'augmentation artificielle sur les données.

Chacune des étapes d'**hicream** et, en particulier, l'analyse différentielle basée sur les pixels ainsi que le clustering, peuvent être modifiées par l'utilisatrice afin d'utiliser une méthode qu'elle jugerait plus pertinente pour ses données et l'objectif de son analyse. Par ailleurs, l'utilisation de l'inférence *post hoc* sur ce type de données de génomique 3D est également une nouveauté.

De multiples améliorations de la méthode peuvent être envisagées, que ce soit d'un point de vue méthodologique (choix de la méthode de clustering, choix de la borne *post hoc*, etc.) qu'en termes d'interprétabilité des résultats au travers de leur visualisation ou de la caractérisation des clusters identifiés.

Choix des régions candidates. La méthode **hicream** permet à l'utilisatrice de donner en entrée n'importe quels clusters de pixels. Dans le cas le plus commun, où elle souhaiterait explorer les résultats et « découvrir » des structures différentielles dans le jeu de données, il est proposé d'utiliser un clustering de Ward avec contrainte de contiguïté sur les pixels.

Actuellement, le choix automatique du nombre de clusters est implémenté dans **hicream** par une recherche de coude sur le critère d'agrégation (variance de l'inertie intra-classe globale). Il est également possible de définir le nombre de clusters souhaités *a priori*. Cependant, les résultats de l'application de la méthode à des jeux de données réels ont montré que ce choix pouvait ne pas être optimal en termes d'interprétabilité et de caractérisation des clusters. En effet, on retrouve par exemple des clusters différentiels proches de la diagonale qui sont contigus et similaires en termes de log-fold change et de p-valeurs ajustées. Ce type de résultat laisse à penser que ces différents clusters n'en forment qu'un

seul à un autre niveau de la hiérarchie. Il pourrait alors s'avérer plus interprétable de s'intéresser à ce plus gros cluster.

Également, dans le contexte où l'utilisatrice chercherait à identifier des clusters différentiels correspondant à une certaine structure donc arborant une forme spécifique, l'information de forme pourrait devenir un critère dans le clustering. Supposons que l'on veuille identifier des clusters qui correspondent à des TADs différentiels et que l'on possède un *triangle-shape score* pour chacun d'entre eux. On pourrait imaginer parcourir la hiérarchie de manière asymétrique et continuer à « descendre » une branche tant qu'on ne trouve pas de cluster qui satisfait le critère de forme avec un TDP intéressant et une cohérence en log-fold change. Ces différentes remarques soulignent l'éventuelle possibilité d'ajouter de nouveaux critères de sélection du clustering, que ce soit en découpant la hiérarchie à un autre niveau que ce qui est proposé actuellement, ou en décidant de la parcourir de manière asymétrique afin d'obtenir les clusters les plus intéressants relativement à des critères de forme ou de taille par exemple.

D'autre part, on pourrait également imaginer changer la contrainte de contiguïté actuellement prise en compte dans le clustering ou bien utiliser une nouvelle méthode de clustering. En particulier, de telles méthodes pourraient viser à identifier des clusters répondant à des contraintes particulières en termes de forme ou de structure. À titre d'exemple, si l'on souhaite identifier des compartiments différentiels, on pourrait imaginer utiliser une normalisation qui permette de mettre en évidence ce type de structure sur les matrices comme la normalisation par rapport à la distance « *observed/expected* » puis appliquer un clustering qui formerait des clusters correspondant à des compartiments.

En somme, cette approche laisse toute latitude aux utilisatrices de la méthode pour définir les régions candidates de quelque manière qu'elles le souhaitent et y incorporer des connaissances *a priori* ou des informations extérieures. Toutes ces perspectives sont envisageables car la garantie statistique obtenue sur chaque cluster par l'inférence *post hoc* n'est pas invalidée par le fait de regarder de nouveaux clusters.

Choix de la borne *post hoc*. Concernant l'inférence *post hoc*, des améliorations spécifiques au type de données traité sont envisageables. En effet, la borne actuellement utilisée – la borne de Simes – peut être conservatrice pour des cas où les p-valeurs ne sont pas indépendantes [Blanchard et al., 2020], comme c'est le cas ici. Des calibrations permettant d'obtenir des bornes plus précises ont d'ores et déjà été proposées dans Blanchard et al. [2020] et appliquées par exemple à l'identification de groupes de gènes différentiels [Enjalbert-Courrech and Neuvial, 2022]. Ces calibrations nécessitent de connaître la distribution des p-valeurs sous l'hypothèse nulle, ce qui peut être obtenu par permutations. Cependant, ce type d'approche ne semble pas directement applicable aux données Hi-C en raison de la difficulté à obtenir des données sous H_0. En effet, nous ne disposons généralement que de quelques réplicats (entre deux et quatre) dans chaque condition, ne permettant pas de réaliser des permutations. De futurs travaux visant à produire des matrices Hi-C sous H_0 par simulations pourraient peut-être permettre de répondre à cette problématique. Cela rejoint les remarques faites précédemment concernant la simulation de données Hi-C pour le benchmark de méthodes.

Visualisation et caractérisation des résultats. La nouveauté de la méthode **hicream**, relativement à l'identification de clusters de formes arbitraires, a nécessité de s'intéresser à une nouvelle manière de représenter ses résultats. En effet, les outils de visualisation existants permettent généralement de représenter une matrice Hi-C, de se focaliser sur certaines parties de celle-ci, éventuellement de représenter des informations biologiques complémentaires telles que des données ChIP-Seq, mais ne permettent pas de représenter des régions délimitées dans la matrice. Pour parcourir les résultats d'**hicream**, il est primordial de pouvoir distinguer les clusters les uns des autres à travers une représentation de leurs frontières ainsi que de pouvoir les caractériser relativement à leur TDP minimal ainsi que les p-valeurs ajustées et le log-fold change des pixels qu'ils contiennent. Le développement d'un outil interactif de visualisation des résultats grâce au package **animint2** a permis de répondre à cette problématique. Néanmoins, de nombreuses améliorations pourraient être implémentées afin de pouvoir par exemple retrouver automatiquement un cluster et sa position en fonction du numéro qui lui est attribué ou encore de proposer de visualiser, pour un même chromosome, des résultats issus de clusterings différents.

Par ailleurs, même si cela pose de nouveaux défis techniques relatifs à la puissance de calcul en lien avec la taille des données, on pourrait imaginer produire une visualisation interactive sur laquelle l'utilisatrice pourrait sélectionner une région de la matrice Hi-C et pour laquelle elle obtiendrait le TDP minimal correspondant. Un tel outil (https://shiny-iidea-sanssouci.apps.math.cnrs.fr/) a déjà été produit pour de l'analyse d'expression différentielle utilisant de l'inférence *post hoc*. Notons que ce type de visualisation requiert de faire des calculs de manière instantanée et peut donc s'avérer très coûteuse tandis que la visualisation que nous proposons actuellement s'appuie sur des tableaux de données construits au préalable et ne réalise aucun calcul.

Au-delà de la visualisation, qui est essentielle et permet d'avoir un premier aperçu de la forme et la position des clusters différentiels dans la matrice, il est intéressant de pouvoir les caractériser plus finement. De ce fait, on pourrait imaginer enrichir l'implémentation du package **hicream** en proposant une fonction de caractérisation des clusters différentiels avec des critères/scores de forme (*triangle-shape score*, *stripe-shape score*, *square-shape score*, etc.) permettant de guider l'utilisatrice pour l'interprétation de ces résultats et d'éventuelles analyses et validations complémentaires.

Intérêt de l'approche interdisciplinaire. L'essentiel de ce travail a consisté à proposer des outils statistiques adaptés pour traiter et analyser des données Hi-C. Ces données biologiques, qui sont issues de manipulations produisant des fragments d'ADN, sont traitées par des pipelines bio-informatiques afin de passer de paires de lectures à des matrices Hi-C, stockées ensuite dans des formats optimisés pour leur analyse. Des outils à l'interface entre bio-informatique et mathématiques permettent de normaliser ces données afin d'enlever les biais techniques et biologiques qu'elles contiennent. Enfin, l'utilisation d'outils statistiques permet de réaliser, notamment, des analyses différentielles de ces données. C'est donc la mise en commun d'outils propres à différents domaines qui permet de résoudre des problématiques complexes. Le travail proposé ici pourra être utilisé par des biologistes et bio-informaticiennes afin d'analyser leurs données mais cet apport de connaissances entre disciplines n'est pas unilatéral. En effet, les contraintes induites

par la nature des données, leur grande dimension, la complexité des structures de dépendance qu'elles présentent sont des éléments qui font émerger des questions de recherche qui poussent à développer et fournir de nouveaux outils statistiques. Au-delà du développement de nouveaux outils et comme illustré ici pour l'inférence *post hoc*, je me permets de formuler l'hypothèse que, pour beaucoup de problématiques relatives à des données biologiques complexes, il existe déjà des apports théoriques permettant d'y répondre. Il revient alors à la biostatisticienne de faire la traduction d'un besoin formulé sur les données biologiques en une problématique théorique puis d'identifier les outils adaptés pour les analyser, potentiellement jamais appliqués dans le contexte considéré. Il reste souvent difficile de faire ce lien et c'est d'autant plus important que les mathématiciennes, statisticiennes, physiciennes et autres scientifiques développant des méthodologies d'analyse proposent des implémentations de leurs méthodes afin d'en faciliter l'utilisation et la valorisation.

Annexes

A Annexes du chapitre 2

Supplementary information for "A comprehensive review and benchmark of differential analysis tools for Hi-C data"

Élise Jorge, Sylvain Foissac, Pierre Neuvial, Matthias Zytnicki, and Nathalie Vialaneix

Contents

1 Supplementary tables **1**

2 Supplementary figures **2**

3 Supplementary information on documentation and installation of tools **6**
3.1 Documentation 6
3.2 Bugs identified during tool installation 6

4 Supplementary methods **7**
4.1 Hi-C data processing of ENCODE human data 7
4.2 Hi-C data processing of CTCF data 7

1 Supplementary tables

Tool	Filter type	Mandatory?
ACCOST	mappability threshold	yes (but can be bypassed using a threshold of 0 or by setting mappability to 1)
CHESS	mappability threshold	no
diffHic	total count user-specified threshold	no
	total count data-driven threshold at matrix level	no
	total count data-driven threshold at bin distance level	no
	remove diagonal bin pairs	no
	select only peaks	no
HiCcompare	user-specified	no
HiCDCPlus	filters out bin pairs with a non significant effect according to distance, GC, and mappability	yes (but not all effects are mandatory to consider)
HOMER	total count data-driven threshold at matrix level	no
multiHiCcompare	total count user-specified threshold	no
Selfish	total count user-specified threshold	no
	remove bin pairs for which genomic distance is above a certain threshold	no

Table S1: Types of filters available in the different tools.

Tool	Normalization type	Performed with?	Mandatory?
ACCOST	normalization of total counts between bins	user-provided bin-specific correction values	yes
	genomic distance effect correction	data-driven distance-specific correction value	yes
CHESS	genomic distance effect correction	observed over expected	yes
diffHic	normalization of matrix sequencing depth	MA correction	no (but highly recommended)
	normalization of total counts between bins	ICE	no
	other	CNV correction	no
FIND	normalization of total counts between bins	VC, VC_SQRT, KR	no
HiCcompare	normalization of matrix sequencing depth	MD correction	no (but highly recommended)
HiCDCPlus	genomic distance effect correction	data-driven distance-specific correction value	yes
HOMER	normalization of total counts between bins	ICE	yes
multiHiCcompare	normalization of matrix sequencing depth	MD correction	no (but highly recommended)
Selfish	normalization of total counts between bins	user-provided bin-specific correction values	no
	normalization of total counts between bins	KR	no
sslHiC	normalization of matrix sequencing depth	TSS	yes
	normalization of matrix sequencing depth	min/max normalization on $\log_{10}$-transformation	yes

Table S2: Types of normalization performed by the different tools.

Tool	Reference	Tool URL	Used version
diffHic	[5]	`https://www.bioconductor.org/packages/release/bioc/html/diffHic.html`	1.26.0
FIND	[2]	`https://bitbucket.org/nadhir/find`	1.0.0
HiCcompare	[10]	`https://www.bioconductor.org/packages/release/bioc/html/HiCcompare.html`	1.16.0
HiCDCPlus	[8]	`https://bioconductor.org/packages/release/bioc/html/HiCDCPlus.html`	1.2.1
multiHiCcompare	[9]	`https://www.bioconductor.org/packages/release/bioc/html/multiHiCcompare.html`	1.12.0
Selfish	[1]	`https://github.com/ay-lab/selfish`	as of 2022/09/01
sslHic	[4]	`https://github.com/lihan97/sslHiC`	as of 2023/04/11

Table S3: Tested tools, reference, source and number of the used version.

2 Supplementary figures

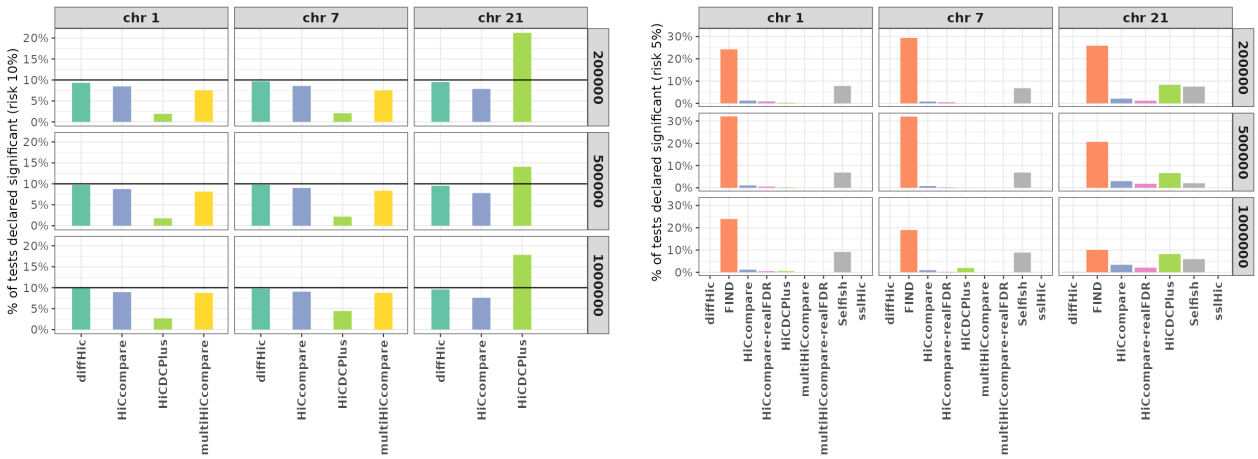


Figure S1: **"H_0 setting".** Percentage of tests declared significant at risk 10% based on p-values (left) and adjusted p-values (right). The black horizontal line (left) indicates the risk level that raw p-values are supposed to control. **sslHiC** could only be used on 500 kb resolution data and **multiHiCcompare** performed no test on chromosome 21 because of its filtering step.

Tool	Option name	Description	Default value	Used value
diffHic	Default values were used. We did not perform filtering based on `aveLogCPM` but performed the trended filter as recommended in the described analysis pipeline.			
FIND	`qvalue`	q-value threshold for exported results	0.001	1
	`chunkSize`	no documentation but used to split the computation into small chunks	50	8
	`nbProcessor`	no documentation but used for multicore processing	2	1
HiCcompare	Default values were used.			
HiCDCPlus	Default values were used.			
multiHiCcompare	Default values were used.			
selfish	Default values were used.			
sslHiC	Default values were used.			

Table S4: Options set to run the different tools. For all other parameters, we used the default values.

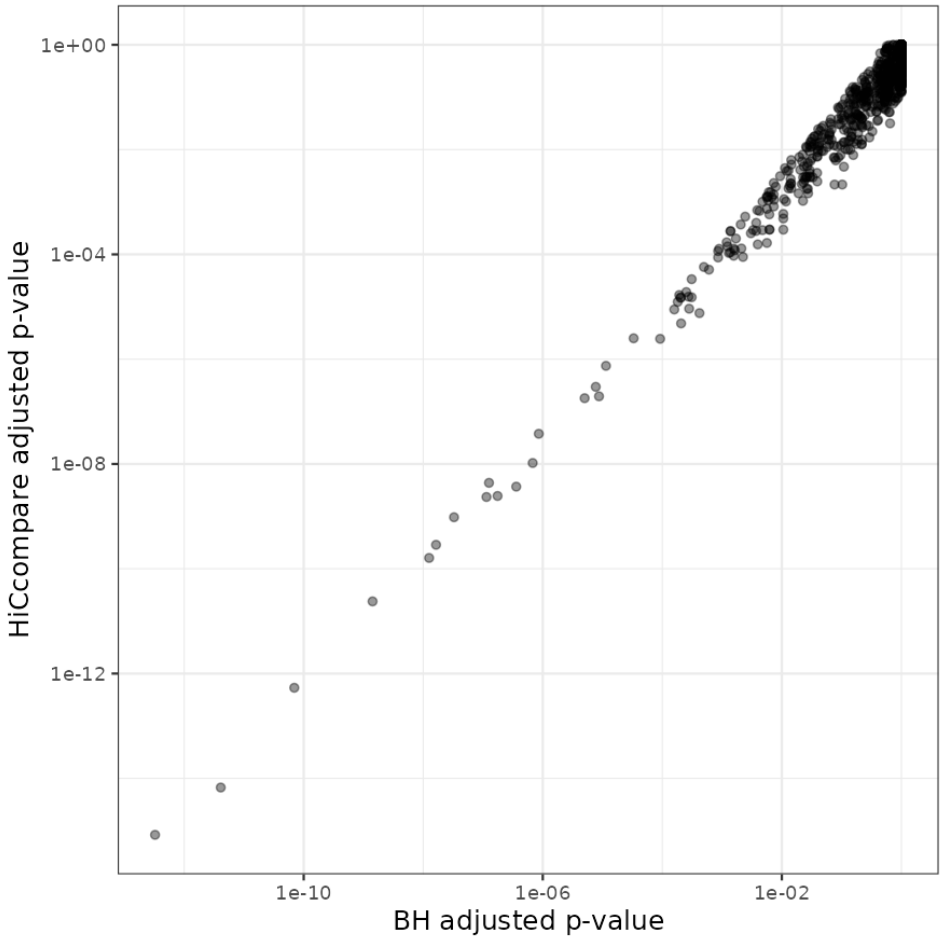


Figure S2: **"H_0 setting".** Adjusted p-values as implemented in the R package **HiCcompare** versus standard BH adjusted p-values. This plot corresponds to results obtained on chromosome 21 at 1 Mb resolution and axes have been $\log_{10}$-scaled.

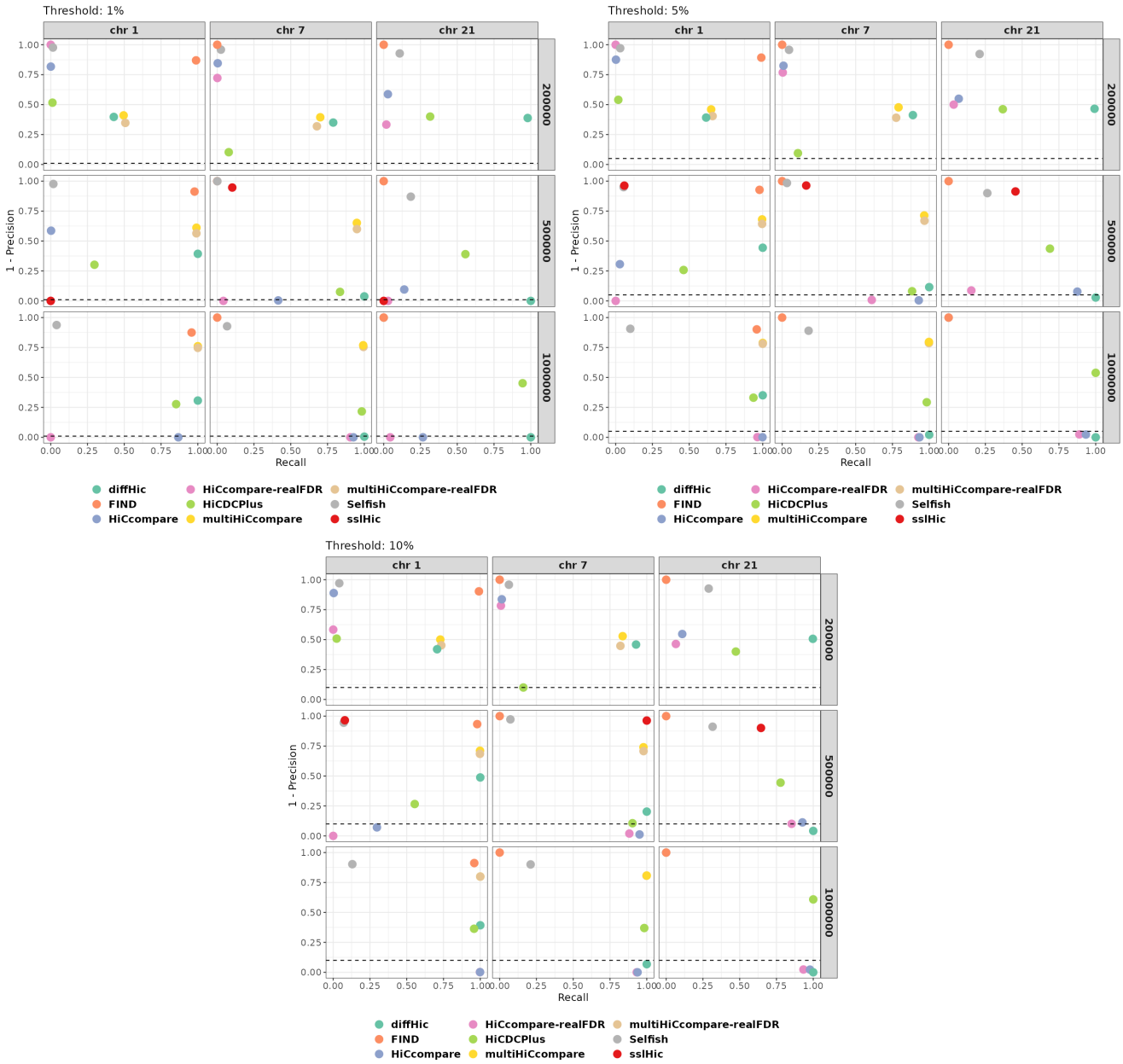


Figure S3: **"H_1 setting".** 1− Precision versus Recall at different thresholds of the adjusted p-values (1%, 5%, and 10%). The dashed horizontal line shows the target FDR that the test should control. **sslHiC** could only be used on 500 kb resolution data and **multiHiCcompare** performed no test on chromosome 21 because of its filtering step.

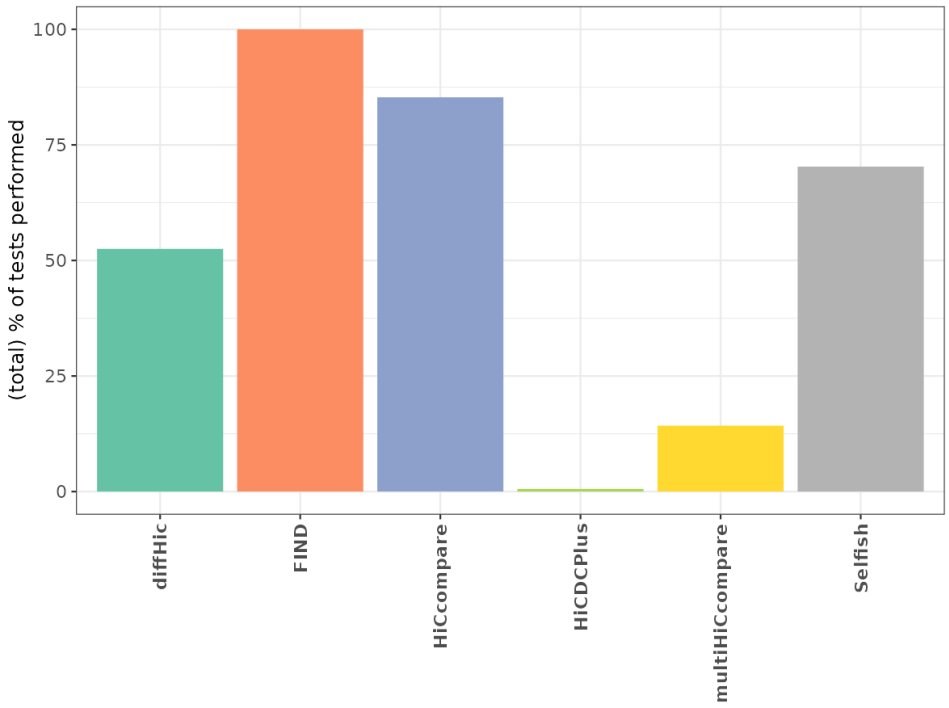


Figure S4: **CTCF depletion dataset.** Percentage of tests performed by each tool (after filtering steps) compared to the original number of bin pairs available in data. **sslHiC** could not be used.

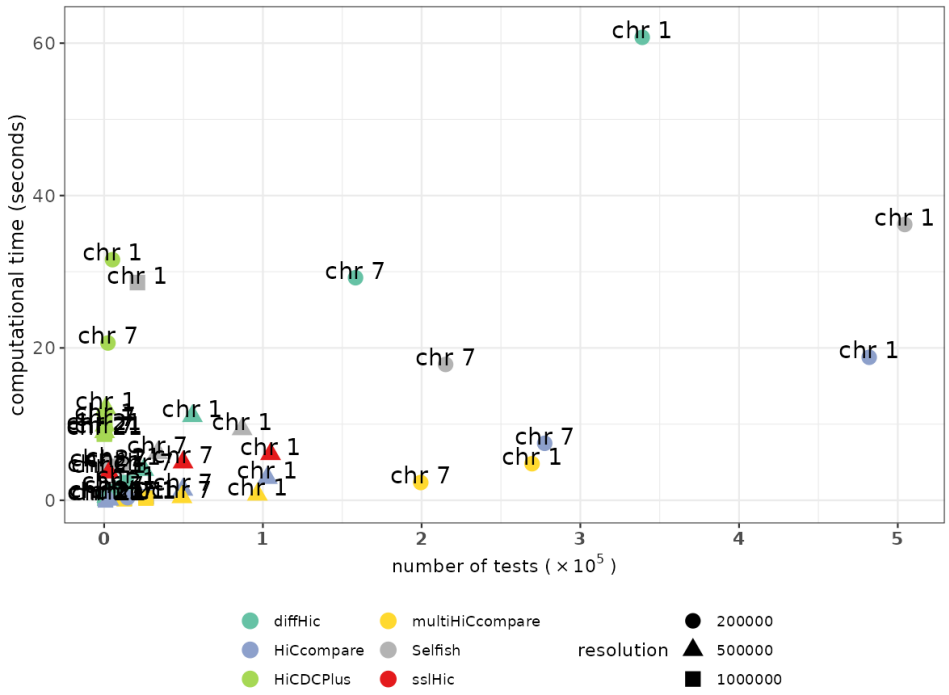


Figure S5: **"H_1 setting".** Computational time with respect to performed number of tests (remaining after the filtering step). **FIND** computational time is not provided because it was too large compare to the other tools to not compromise the plot readability. **sslHiC** could only be used on 500 kb resolution data and **multiHiCcompare** performed no test on chromosome 21 because of its filtering step.

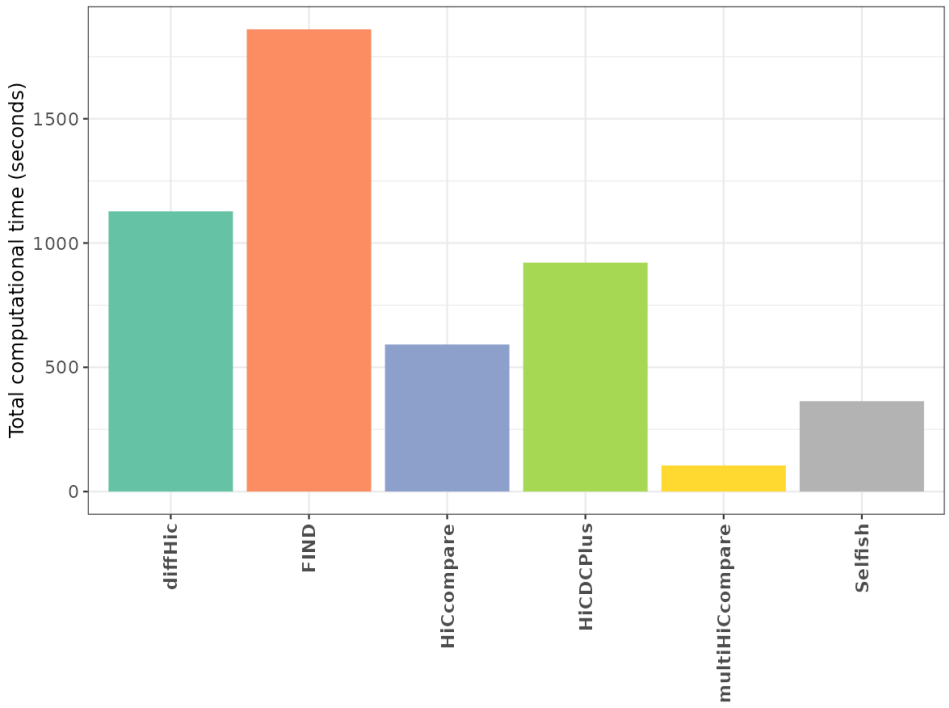


Figure S6: **CTCF depletion dataset.** Total computational time required by each tool. **sslHiC** could not be used.

3 Supplementary information on documentation and installation of tools

3.1 Documentation

The documentation is of varying precision: Bioconductor R packages (**diffHic**, **HiCcompare**, **HiCDCPlus**, and **multiHiCcompare**) are particularly easy to use thanks to their extensive documentation. On the contrary, **FIND** documentation is incomplete and imprecise: For instance, only some of the exported functions are documented (*e.g.*, among the methods to extract differential analysis results, `getDCIs_XXX`, only the one for matrix lists is documented whereas the method designed for the outputs of the import function `loadHicExperiment` is not documented), the documentation of some arguments is not clear (*e.g.*, "qvalue: numeric. cutoff q-value. Taken into considiration i", in which the sentence is cut midway), and the documentation of some other arguments is missing (*e.g.*, `isrOP_qval`). In addition, **FIND** includes a vignette but this vignette is not automatically built during the installation (it has dependencies to packages that are not listed in the "required packages" section of the documentation). When we tried to build it with **knitr** after installing the tool, it yielded to a segmentation fault of undefined origin.

Selfish and **sslHiC** offer a basic documentation in the `README` file of their code repository, with a description of all the function parameters and a basic example. **ACCOST** has approximately the same level of documentation, included in a HTML manual in their code repository (note that this manual can not be read online and has to be downloaded to be used). In contrast, **CHESS** provides a complete "Read the Docs" documentation at `https://chess-hic.readthedocs.io`, with installation instruction, examples, parameter description, and some practical advices on how to set these parameters with your own data.

Finally, **HOMER** is a set of Perl scripts, with some dependencies to external R packages (as **edgeR**, [7]). The installation instructions are detailed though not easy to follow for a basic user (Windows users are instructed to emulate a linux environment to use **HOMER**). An installation script is provided to download a ZIP file with the source code (that does not seem to be available in a standard code repository). The documentation is also provided on the tool website `http://homer.ucsd.edu/homer/`. It is detailed but not easy to navigate and finding the documentation related to differential analysis `http://homer.ucsd.edu/homer/interactions/HiCinteractions.html` might require some efforts.

3.2 Bugs identified during tool installation

Some tools could not be installed on our infrastructure just following the installation instructions, and some of the problems most likely were caused by a bug in the tool. When we thought we found a bug, we filed an issue to the code repository:

- **sslHiC**
  - `https://github.com/lihan97/sslHiC/issues/2`
  - `https://github.com/lihan97/sslHiC/issues/3`

- **selfish**
  - `https://github.com/ay-lab/selfish/issues/26`

HOMER does not include an issue tracker and an email was used instead.

4 Supplementary methods

4.1 Hi-C data processing of ENCODE human data

To perform the simulations for the H_0 and H_1 settings, we used a human Hi-C dataset generated by the ENCODE consortium [11]. Raw reads from five sequencing runs of the same Hi-C library were downloaded in fastq format from the ENCODE data portal [6] at `https://www.encodeproject.org/` using the experiment accession ID ENCSR295BDK. This library comes from a colon sample of a 37-year old male donor, each sequencing run providing a technical replicate.

To obtain a Hi-C matrix for each replicate, raw sequencing reads were processed using the `nf-core/hic` pipeline [3] v1.2.2 on the assembly version GRCh38 of the human genome using the following arguments: `--min_mapq 10 --restriction_site '^GATC' --ligation_site 'GATCGATC' --min_insert_size 20 --max_insert_size 1000 --rm_singleton --rm_dup --skip_ice --bwt2_opts_end2end '--very-sensitive -L 30 --score-min L,-0.6,-0.2 --end-to-end --reorder' --bwt2_opts_trimmed '--very-sensitive -L 20 --score-min L,-0.6,-0.2 --end-to-end --reorder'`. Matrix visualizations were generated using the `hicPlotMatrix` function of `HiCExplorer` v3.7.2 with the parameters `--clearMaskedBins --log1 --colorMap OrRd --perChromosome --dpi 300 --fontsize 16 --rotationX 45`.

4.2 Hi-C data processing of CTCF data

To assess the biological relevance of the differential analysis results we used a large scale dataset from an existing study on the role of CTCF during cell cycle progression in mouse [13]. This study features an erythroblasts cell line that has been modified to integrate an auxin-induced depletion system that targets the CTCF protein concentration. In particular, Hi-C has been performed for three replicates of two conditions: a wild-type condition, with no CTCF depletion ("CTCF+"), and a CTCF depleted condition ("CTCF-"), both at the post-mitotic G1 phase, 120 minutes after cell cycle reentry. Raw interaction counts of the six Hi-C matrices were directly downloaded from the GEO platform uwing the accession ID "GSE168251" from `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE168251`. File list: `GSM5133388_120min_no_a_rep1.cool.gz`, `GSM5133389_120min_no_a_rep2.cool.gz`, `GSM5133390_120min_no_a_rep3.cool.gz`, `GSM5133391_120min_with_a_rep1.cool.gz`, `GSM5133392_120min_with_a_rep2.cool.gz`, and `GSM5133393_120min_with_a_rep3.cool.gz`. Matrices in cool format were exported using the HiC-Pro text format by chromosome at 100 kb resolution. Obtained files were used as input for differential analysis with each tested tools using the "CTCF+" vs. "CTCF-" contrast. Resulting differences in interaction values (adjusted p-value $< 5\%$) were kept for further validation.

In order to assess the biological relevance of the resulting differential interactions according to each tested tool, in the absence of ground truth, we used an existing dataset of genome-wide CTCF occupancy that was obtained by ChIP-Seq experiment on the same murine cell line [12]. Resulting CTCF binding sites were directly downloaded in BED format from the GEO website using the accession ID GSE129997 (url: `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE129997`). File name: `GSE129997_CTCF_midG1.narrowPeak.gz`.

CTCF occupancy data from ChIP-seq experiments and differential analysis results from each tool were integrated by projecting them on the `mm9` version of the mouse genome at the 100 kb bin level resolution using the **bedtools intersect -c** command. Each genomic bin was assigned two values: the number of differential interactions that involved the bin and the number of detected CTCF binding sites. Spearman correlation between these numbers was then computed across all bins using the `cor()` R function with the `method = "spearman"` parameter.

References

[1] Abbas Roayaei Ardakany, Ferhat Ay, and Stefano Lonardi. Selfish: discovery of differential chromatin interactions via a self-similarity measure. *Bioinformatics*, 35(14):i145–i153, 2019.

[2] Mohamed Nadhir Djekidel, Yang Chen, and Michael Q. Zhang. FIND: difFerential chromatin INteractions Detection using a spatial Poisson process. *Genome Research*, 28:412–422, 2018.

[3] Philip A. Ewels, Alexander Peltzer, Sven Fillinger, Harshil Patel, Johannes Alneberg, Andreas Wilm, Maxime Ulysse Garcia, Paolo Di Tommaso, and Sven Nahnsen. The nf-core framework for community-curated bioinformatics pipelines. *Nature Biotechnology*, 38:276–278, 2020.

[4] Han Li, Xuan He, Lawrence Kurowski, Ruotian Zhang, Dan Zhao, and Jianyang Zeng. Improving comparative analyses of Hi-C data via contrastive self-supervised learning. *Briefings in Bioinformatics*, 24(4):bbad193, 2023.

[5] Aaron T.L. Lun and Gordon K. Smyth. diffHic: a Bioconductor package to detect differential genomic interactions in Hi-C data. *BMC Bioinformatics*, 16:258, 2015.

[6] Yunhai Luo, Benjamin C Hitz, Idan Gabdank, Jason A Hilton, Meenakshi S Kagda, Bonita Lam, Zachary Myers, Paul Sud, Jennifer Jou, Khine Lin, Ulugbek K. Baymuradov, Keenan Graham, Casey Litton, Stuart R. Miyasato, J. Seth Strattan, Otto Jolank, Jin-Wook Lee, Forrest Y. Tanaka, Philipp Adenekan, Emma O'Neill, and J. Michael Cherry. New developments on the Encyclopedia of DNA Elements (ENCODE) data portal. *Nucleic Acids Research*, 48(D1):D882–D889, 2020.

[7] Mark D. Robinson, David J. McCarthy, and Gordon K. Smyth. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010.

[8] Merve Sahin, Wilfred Wong, Yingqian Zhan, Kinsey Van Deynze, Richard Koche, and Christina S. Leslie. HiC-DC+ enables systematic 3D interaction calls and differential analysis for Hi-C and HiChIP. *Nature Communications*, 12(1):3366, 2021.

[9] John C. Stansfield, Kellen G. Cresswell, and Mikhail G. Dozmorov. multiHiCcompare: joint normalization and comparative analysis of complex Hi-C experiments. *Bioinformatics*, 35(17):2916–2923, 2019.

[10] John C. Stansfield, Kellen G. Cresswell, Vladimir I. Vladimirov, and Mikhail G. Dozmorov. HiCcompare: an R-package for joint normalization and comparison of HI-C datasets. *BMC Bioinformatics*, 19:279, 2018.

[11] The ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, 2012.

[12] Haoyue Zhang, Daniel J. Emerson, Thomas G. Gilgenast, Katelyn R. Titus, Yemin Lan, Peng Huang, Di Zhang, Hongxin Wang, Cheryl A. Keller, Belinda Giardine, Ross C. Hardison, Jennifer E. Phillips-Cremins, and Gerd A. Blobel. Chromatin structure dynamics during the mitosis-to-G1 phase transition. *Nature*, 576(7785):158–162, 2019.

[13] Haoyue Zhang, Jessica Lam, Di Zhang, Yemin Lan, Marit W. Vermunt, Cheryl A. Keller, Belinda Giardine, Ross C. Hardison, and Gerd A. Blobel. CTCF and transcription influence chromatin structure re-configuration after mitosis. *Nature Communications*, 12:5157, 2021.

B Annexes du chapitre 3

Supplementary information for "hicream: a flexible framework to identify significantly different regions in Hi-C data"

Elise Jorge, Toby Dylan Hocking, Pierre Neuvial, Nathalie Vialaneix, and Sylvain Foissac

Contents

S1 Supplementary methods **2**
S1.1 Pixel-based differential analysis with **diffHic DA** 2
S1.2 Connectivity constrained 2D hierarchical clustering 2
S1.2.1 Variable transformation and normalization 2
S1.2.2 Constraint to ensure unique connected component 2
S1.2.3 Choice of 2D hierarchical clustering algorithm 3

S2 Supplementary information on data processing **4**
S2.1 Semi-synthetic dataset 4
S2.2 CTCF depletion study 4
S2.3 Neuronal differentiation study 4
S2.4 Analysis of cluster shapes 5
S2.5 Comparison with **TADCompare** and **dcHiC** results 6
S2.6 CTCF enrichment 6

S3 Expected impact of CTCF depletion **7**

S4 Supplementary figures **8**

S1 Supplementary methods

S1.1 Pixel-based differential analysis with diffHic DA

In **diffHic** differential analysis, a normalization step is performed using the correction of the MA plot with Locally Estimated Scatterplot Smoothing (LOESS) [Ballman et al., 2004] as implemented in the R package **csaw** [Lun and Smyth, 2016]. The statistical tests then rely on generalized linear models (GLM) based on a Negative Binomial (NB) distribution assumption for the pixel counts (as implemented in the R package **edgeR** [Robinson et al., 2010]). For each pixel (i, j) such that $i \leq j$, a log-linear model

$$\log \lambda_{ij}^{g,l} = \beta_{ij}^{(0)} + \beta_{ij}^{(1)} \mathbb{1}_{\{g=2\}} + \log N_{g,l},$$

is fitted, where for sample l in condition g, $\lambda_{ij}^{g,l} = \mathbb{E}(m_{ij}^{g,l})$ is the pixel-specific mean of the NB distribution, and $N_{g,l}$ is an estimate of the library size. The statistical test of the null hypothesis

$$\mathrm{H}_{0,ij}\text{: } \text{“}\beta_{ij}^{(1)} = 0\text{”}$$

for every pixel (i, j) results in a p-value half-matrix, $(\mathbf{P}_{ij})_{i,\, j=1\ldots,p,\ i\leq j}$, which contains the result of $m \leq \frac{p(p+1)}{2}$ tested hypotheses, where m is the number of pixels for which at least one interaction has been observed in one of the r Hi-C matrices.

S1.2 Connectivity constrained 2D hierarchical clustering

S1.2.1 Variable transformation and normalization

For **hicream diff**, the clustering is performed using two variables: the adjusted p-value and log-fold change computed using **diffHic**. In practice, we first transform the variables into $(-\log_{10} \mathbf{P}^{\mathrm{adj}}, \log_2 \mathrm{FC})$ and then separately normalize them into $[0, 1]$ by applying a min-max normalization. The use of two log transformations with different bases (that correspond to the respective standard bases for these two quantities) is not an issue as the min-max normalization is invariant relative to log-base choice.

S1.2.2 Constraint to ensure unique connected component

The graph $\mathcal{G}$ links pixels that are adjacent in the Hi-C matrix in such a way that a pixel has, at most, 4 neighbors. However, due to missing bins in the data (*e.g*, pixels $q := (i, j)$ that have zero counts for a fixed q in all r matrices), some pixels could have no neighbors, potentially leading to a disconnected graph.

To solve this issue, we extend the definition of connectivity in order to minimize the number of connected components in the graph: $q_1 = (i_1, j_1)$ and $q_2 = (i_2, j_2)$ are connected in $\mathcal{G}$ iif one of the following condition is satisfied:

$$
\begin{aligned}
i_1 = i_2,\ j_2 &= \underset{j^\star > j_1,\, (i_2, j^\star) \in V}{\arg\min} |j^\star - j_1| \\
i_1 = i_2,\ j_2 &= \underset{j^\star < j_1,\, (i_2, j^\star) \in V}{\arg\min} |j^\star - j_1| \\
j_1 = j_2,\ i_2 &= \underset{i^\star > i_1,\, (i^\star, j_2) \in V}{\arg\min} |i^\star - i_1| \\
j_1 = j_2,\ i_2 &= \underset{i^\star < i_1,\, (i^\star, j_2) \in V}{\arg\min} |i^\star - i_1|
\end{aligned}
$$

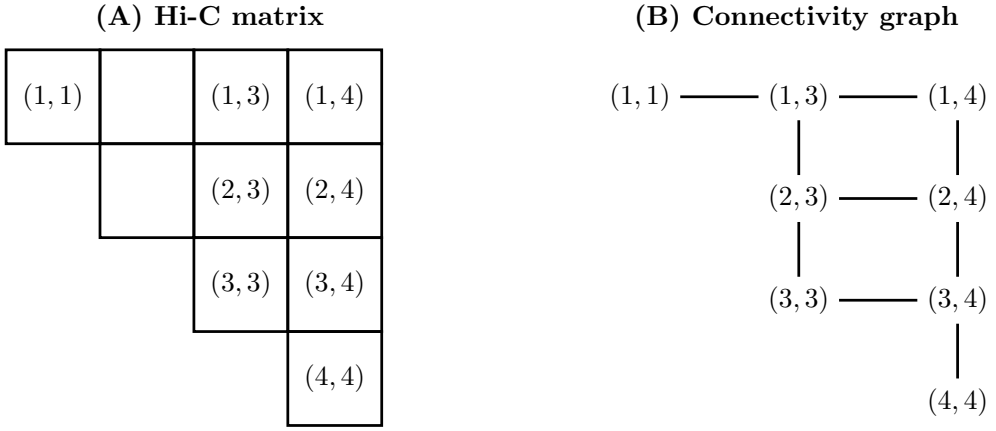


Figure S1: **Illustration of the connectivity constraint.** (A) Extract of the right upper part of a Hi-C matrix where non observed pixels are identified by empty squares (the other pixels are observed). (B) Resulting connectivity graph used in our 2D clustering.

Figure S1 gives an example of the resulting graph: Pixel $(1, 1)$ has no direct contiguous neighbor as the bin 2 is "missing", but, the latter definition ensures that Pixel $(1, 1)$ is connected to the rest of the graph through its closest right pixel, *i.e*, Pixel $(1, 3)$. In practice, this results in a graph that contains only one connected component. However, in some rare cases (e.g., if for a pixel (i, j), there exists no other pixel of the form (i', j) or (i, j')), the graph could contain more than one connected component.

S1.2.3 Choice of 2D hierarchical clustering algorithm

We propose to perform a two-dimensional connectivity-constrained agglomerative clustering. In short, the agglomerative clustering is based on Ward's clustering method [Ward Jr, 1963] and we used the constrained version formally defined in Lebart [1978], Grimm [1987], Gordon [1996]. Here, the constraint is encoded in graph $\mathcal{G}$. This approach was already used by Thirion et al. [2014] to obtain clustering of fMRI pixels arranged in a 3D space. This results in a dendrogram, which can contain a few rever-

sals (due to the contiguity constraint; see Randriamihamison et al. [2021]), from which we can obtain clusters $\{C_1, \cdots, C_K\}$ that form a partition of the pixels by selecting a level at which to cut the hierarchy.

S2 Supplementary information on data processing

S2.1 Semi-synthetic dataset

Semi-synthetic data were obtained using an ENCODE dataset [The ENCODE Project Consortium, 2012]. More precisely, we used the Hi-C experiment performed on a human colon sample (accession number: ENCSR295BDK). The raw sequencing reads were processed using nf-core/hic pipeline [Ewels et al., 2020] v1.2.2 on the assembly version GRCh38 of the human genome in order to obtain Hi-C matrices. Processed Hi-C matrices thus correspond to five technical replicates for three different chromosomes at three different resolutions and are available at `https://doi.org/10.57745/LR0W9R`.

S2.2 CTCF depletion study

We used a large scale dataset from an existing study on the role of CTCF during cell cycle progression in post-mitotic G1E-ER4 mouse cells [Zhang et al., 2021]. The study was conducted under two conditions: either accute depletion of CTCF through an auxin-inducible degron system (CTCF- condition, with three replicates) or in the control condition without auxin-induced depletion (CTCF+ condition, with three replicates). Raw interaction counts of the six Hi-C matrices were directly downloaded from the GEO platform using the accession number GSE168251 from `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE168251` (`GSM5133388_120min_no_a_rep1.cool.gz`, `GSM5133389_120min_no_a_rep2.cool.gz`, `GSM5133390_120min_no_a_rep3.cool.gz`, `GSM5133391_120min_with_a_rep1.cool.gz`, `GSM5133392_120min_with_a_rep2.cool.gz`, and `GSM5133393_120min_with_a_rep3.cool.gz`). Matrices in cool format were exported in HiC-Pro text format by chromosome at 10 kb resolution and coarsened to 200 kb resolution using Bioconductor package **HiContacts**. Resulting files by replicate and chromosome along with the genome-wide index are provided at `https://doi.org/10.57745/IEUSLV`. Corresponding narrow peaks from CTCF ChIP-seq experiments were downloaded from GEO (accession GSE129997). This data corresponds to the CTCF+ condition, *e.g.*, to the expected positions of CTCF sites in a control condition [Zhang et al., 2019].

S2.3 Neuronal differentiation study

We used a large scale dataset from an existing study on neuronal differentiation in mouse [Bonev et al., 2017] and processed the data as described in Maigné et al. [2025]. In brief, sequencing reads were downloaded from the SRA (`https://www.ncbi.nlm.nih.gov/sra`) using the accession ID SRP101791 (GEO: GSE96107) and processed using the nf-core/hic pipeline v1.3.0 on

the assembly version GRCm39 of the mouse genome using the following arguments: `--min_mapq 10 --digestion 'dpnii' --min_insert_size 20 --max_insert_size 1000 --bwt2_opts_end2end '--very-sensitive -L 20 --score-min L,-0.6,-0.3 --end-to-end --reorder' --bwt2_opts_trimmed '-5 5 --very-sensitive -L 20 --score-min L,-0.6,-0.3 --end-to-end --reorder'`. Samples correspond to 2 cellular differentiation stages during neuronal development: mouse embryonic stem cells (ES) and cortical neurons (CN), with 4 replicates per cell type (GEO Accession IDs: GSM2533818, GSM2533819, GSM2533820, and GSM2533821 for ES, and GSM2533826, GSM2533827, GSM2533828, and GSM2533829 for CN). Corresponding ChIP-seq data were obtained from the ReMap ChIP-seq track of the UCSC Table Browser, selecting the CTCF.mESC lines of the ReMapTFs Table.

Processed data are available at `https://doi.org/10.57745/IEUSLV`.

S2.4 Analysis of cluster shapes

Both datasets from the CTCF depletion and the neuronal differentiation studies were analyzed with **hicream** using the **hicream diff** clustering mode.

To characterize the shape of the resulting clusters, three shape-related metrics were computed and assigned to each cluster: a triangle-, a stripe-, and a square-score.

- `triangle_score`: defined as $s/(b^2/2)$, where s is the cluster size (number of pixels) and b the total number of distinct genomic bins involved in the cluster, computed as the union of all row and column bin indices across its contained pixels. This score reaches its maximum when the cluster resembles a topologically associating domain (TAD), i.e. a triangular structure along the diagonal of the contact matrix.

- `stripe_score`: defined as $\log(p_i/p_j)$, where $p_i = i_{\max} - i_{\min} + 1$ and $p_j = j_{\max} - j_{\min} + 1$ are the genomic projections of the cluster along its row and column axes, respectively. A value of 0 is expected for radially symmetric clusters, whereas deviations from 0 indicate elongation along one axis. The sign encodes the orientation of the elongation, while the absolute value is used for the subsequent PCA.

- `square_score`: defined as s/P, where s is the number of pixels in the cluster and P its perimeter, i.e. the total number of pixel sides not shared with any neighboring pixel belonging to the same cluster. Given that each pixel occupies a unit square, this score is maximized when the cluster approximates a square geometry.

These three shape-related metrics, along with the TDP and the absolute log-fold change (logFC) were then used to perform a Principal Component Analysis (PCA) on the set of differential clusters for each study.

S2.5 Comparison with TADCompare and dcHiC results

hicream results obtained for the neuronal differentiation study were compared with **TADCompare** and **dcHiC** results.

TADCompare was run on all chromosomes of the Hi-C data from the neuronal differentiation study. Since the method does not allow for replicates, the replicates were previously merged by summing the Hi-C matrices from the same condition. Genomic regions corresponding to significantly shifted TADs enriched in CN (resp. ES) counts were then extracted.

Selected clusters were first projected onto the genome to identify the corresponding genomic boundaries and their flanking bins. The boundaries were defined as the first and last genomic bins among all interacting bins within each cluster. We then tested for the enrichment in bins identified as "significantly shifted" by **TADCompare** at the boundaries compared with the flanking bins. Specifically, we computed the number of such bins falling in the boundary bins versus those falling in the flanking bins and we compared the resulting distributions using a paired Wilcoxon test.

dcHiC was run on all chromosomes of the Hi-C data from the neuronal differentiation study. Pixels (i, j) from every chromosome were then described with:

- the TDP of the cluster they belong to;
- the type of compartments the two corresponding bins belong to in both conditions, as obtained from **dcHiC**. We encoded this information using the notation $\mathrm{X}_i^{\mathrm{ES}}\mathrm{X}_j^{\mathrm{ES}}2\mathrm{X}_i^{\mathrm{CN}}\mathrm{X}_j^{\mathrm{CN}}$ where $X_\ell^k \in \{\mathrm{A}, \mathrm{B}\}$ is the compartment of bin ℓ (i or j) in condition k (ES or CN). For instance, "AA2BB" describes a pixel between two bins in "A" compartment in ES condition that becomes a pixel between two bins in "B" compartment in CN.

Distribution of positive TDP with respect to pixel compartment types were then obtained, considering two groups of compartment types: the group characterizing compartment changes being composed of types "AA2BB" and "BB2AA" and the other group being composed of all the other cases. In addition, a Wilcoxon test was performed to compare TDP distributions between the two groups of compartment types.

S2.6 CTCF enrichment

Results obtained on both the CTCF depletion and the neuronal differentiation datasets were compared with CTCF signals. Selected **hicream** clusters were first projected onto the genome to identify the corresponding genomic boundaries and their flanking bins. The boundaries were defined as the first and last genomic bins among all interacting bins within each cluster. We then tested for CTCF enrichment at the boundaries compared with the flanking bins, using the genomic coordinates of active CTCF binding sites identified by ChIP-seq experiments. Specifically, we computed the number of CTCF peaks overlapping the boundary bins versus the flanking bins and compared the resulting distributions using a paired Wilcoxon test.

In addition, the distribution of active CTCF binding sites was further characterized by profiling their relative positions with respect to the cluster boundaries. To do this, for each selected cluster $C \subset \{(i,j)|1 \leq i \leq j \leq p\}$, we identified the two boundaries of its projection on the chromosome, which correspond to the bins $\underline{i}$ and $\overline{j}$ for which we have:

$$\begin{aligned}
&\underline{i} \leq \overline{j} \\
&\exists j^* \text{ s.t, } (\underline{i}, j^*) \in C \text{ and } \forall (i,j) \in C,\ \underline{i} \leq i \\
&\exists i^* \text{ s.t, } (i^*, \overline{j}) \in C \text{ and } \forall (i,j) \in C,\ \overline{j} \geq j.
\end{aligned}$$

Then, the average number of CTCF peaks per kb was computed for 20 genomic intervals:

- 10 genomic intervals of equal sizes that each correspond to 10% of the total length of the genomic interval $[\underline{i}, \overline{j}]$;
- 5 flanking regions of equal sizes at 1Mb left (resp. right) of boundary $\underline{i}$ (resp. $\overline{j}$).

Moreover, we tested all cluster boundaries for statistical enrichment of known transcription factor binding sites (TFBSs) from the JASPAR database (CORE Vertebrates 2024) using the AME tool from the MEME Suite (v5.5.4). The background model for this enrichment analysis was built by extracting the flanking sequences around each cluster in order to account for local biases in base composition.

Last, genes with enriched binding motifs from the TFBS analysis were also subjected to a Gene Ontology functional analysis using the Panther Classification System tool from `https://geneontology.org` (PANTHER GO-Slim Biological Process, Fisher exact test and Bonferroni multiple testing correction).

S3 Expected impact of CTCF depletion

Figure S9 illustrates the expected impact of CTCF depletion on the data, particularly at the TAD level. As CTCF is widely known to be associated with TAD boundary formation, we expect a disruption of these structures in the CTCF- condition. Therefore, for two TADs separated by a boundary in the CTCF+ condition, we expect this boundary to be weakened or deleted in the CTCF- condition. This allows more frequent interactions between two positions located in contiguous TADs and leads to differential clusters of positive log-fold change located within inter-TAD regions. In parallel, we expect less frequent interactions within TADs and thus, intra-TAD differential clusters of negative log-fold change.

In conclusion, after projecting each cluster and representing the average CTCF coverage as described in Section S2.6, two different behaviors are expected whether the clusters have an average log-fold change that is positive or negative. If the average log-fold change is negative, we expect a bi-modal distribution with two peaks in CTCF coverage corresponding to the boundary regions. On the other hand, if the average log-fold change is positive, we expect a unimodal distribution with a peak of CTCF coverage located between the two boundaries.

S4 Supplementary figures

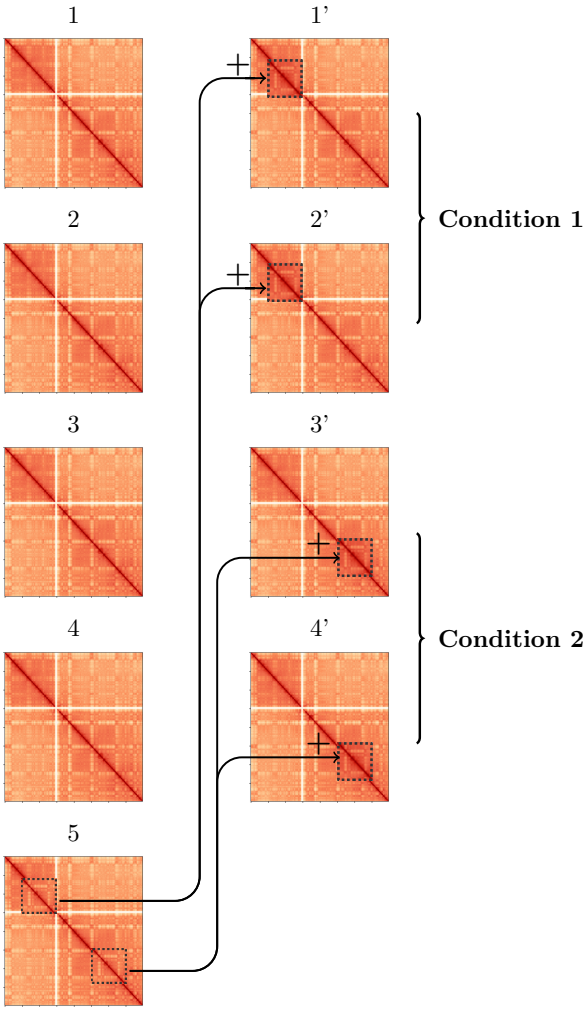


Figure S2: **Semi-synthetic setting.** The original Hi-C matrices consist of five technical replicates from the ENCODE project (left). An artificial Condition 1 was arbitrarily created using Hi-C matrices 1 and 2, to which counts from the fifth matrix were locally added within a predefined target region. This region corresponds to pixels whose interacting bins are located between the 20th and 40th quantiles of chromosome positions. Condition 2 was built using matrices 3 and 4, with counts from a different region of the fifth matrix added. This second region corresponds to pixels whose interacting bins are located between the 60th and 80th quantiles of chromosome positions.

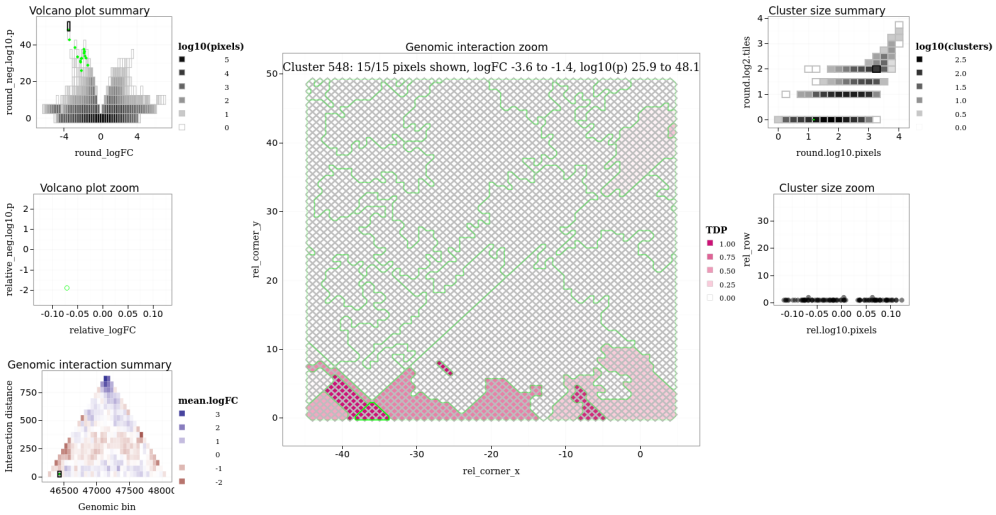
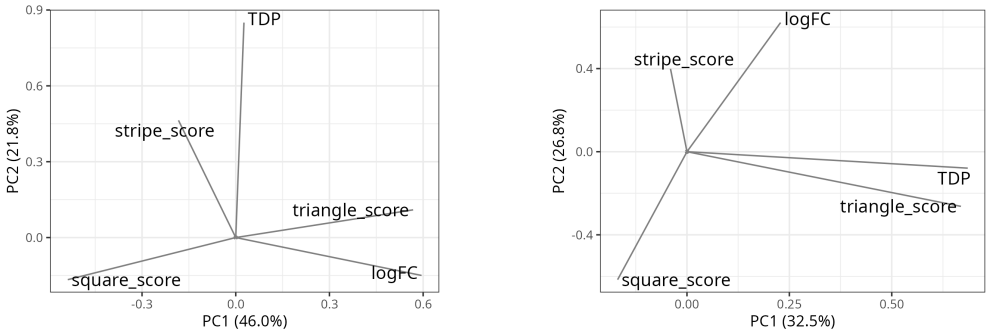


Figure S3: **Neuronal differentiation study: Dynamic visualization of hicream results for chromosome 18 at 50kb resolution with animint2.** The *tile* (genomic region of size 50×50 bins) selected on the *Genomic interaction summary* frame is displayed on the *Genomic interaction zoom* panel. Clusters are outlined in green and each pixel is colored relative to the TDP (True Discovery Proportion) computed using **hicream** for its cluster. Each pixel of the cluster selected on the *Genomic interaction zoom* panel is displayed as a green point on the $\log \mathrm{FC}/\ p$-value volcano plot of *Volcano plot summary*. Clusters can be selected based on the *Volcano plot summary* and *Volcano plot zoom*. It is also possible to select clusters based on their size through the *Cluster size summary* and *Cluster size zoom* panels. Selection of a cluster or tile in one frame automatically updates the other frames.



(a) CTCF depletion study

(b) Neuronal differentiation study

Figure S4: **PCA variable loadings in the CTCF depletion (a) and the neuronal differentiation (b) study.** Principal Component Analysis was perfomed on the differential clusters using their functional (TDP, |logFC|) and shape-related (`triangle_score`, |`stripe_score`|, `square_score`) features. Results show a different association of TDP with the shape-related features: TDP is more associated with `stripe_score` in the CTCF depletion study and with `triangle_score` in the neuronal differentiation study.

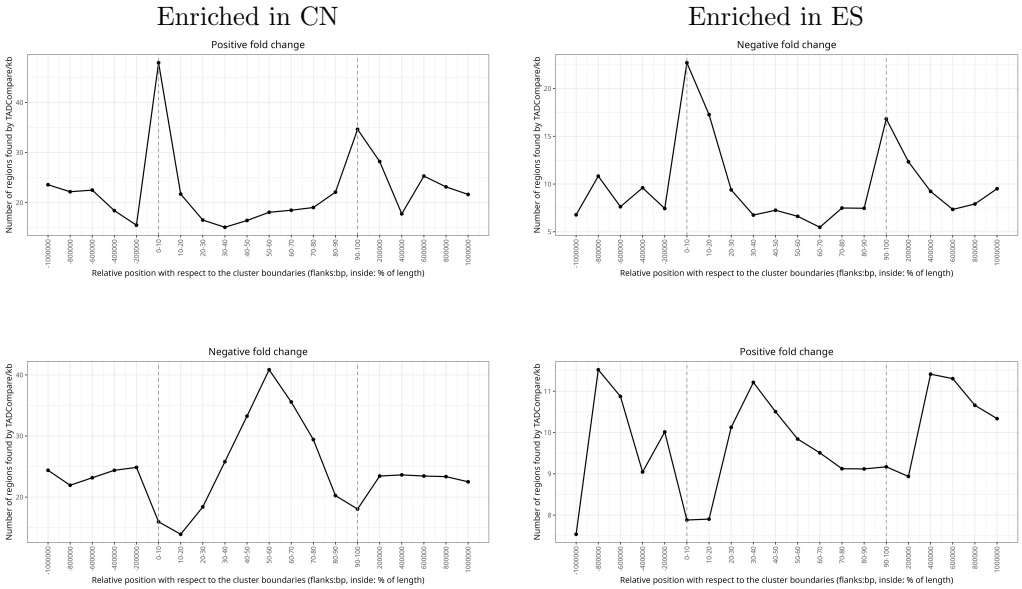


Figure S5: **Comparison between hicream and TADCompare results (neuronal differentiation study).** Enrichment of differentially shifted positions found by **TADCompare** at **hicream** cluster boundaries. Left: Regions enriched in CN for **TADCompare**. Right: Regions enriched in ES for **TADCompare**. First row: **hicream** cluster with logFC sign consistent with **TADCompare** enrichment (*e.g.* positive for CN enrichment and negative for ES enrichment). Second row: **hicream** cluster with logFC sign opposite to **TADCompare** enrichment.

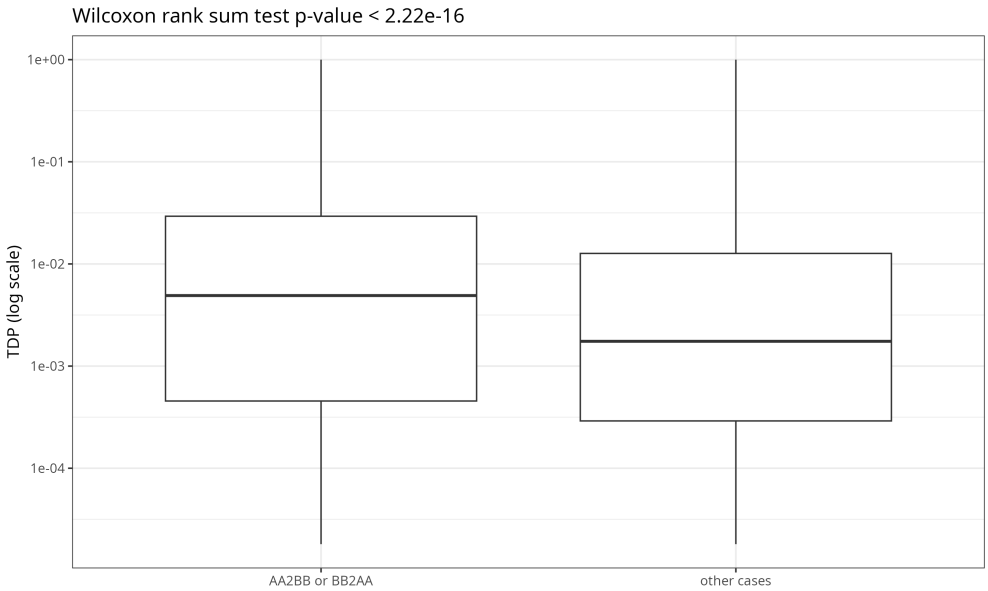


Figure S6: **Comparison between hicream and dcHiC results (neuronal differentiation study).** Distribution (at pixel level) of the pixel cluster TDP with respect to its **dcHiC** status: major compartment change (AA2BB or BB2AA) or other cases. Only pixels in clusters with a positive TDP were considered for this analysis.

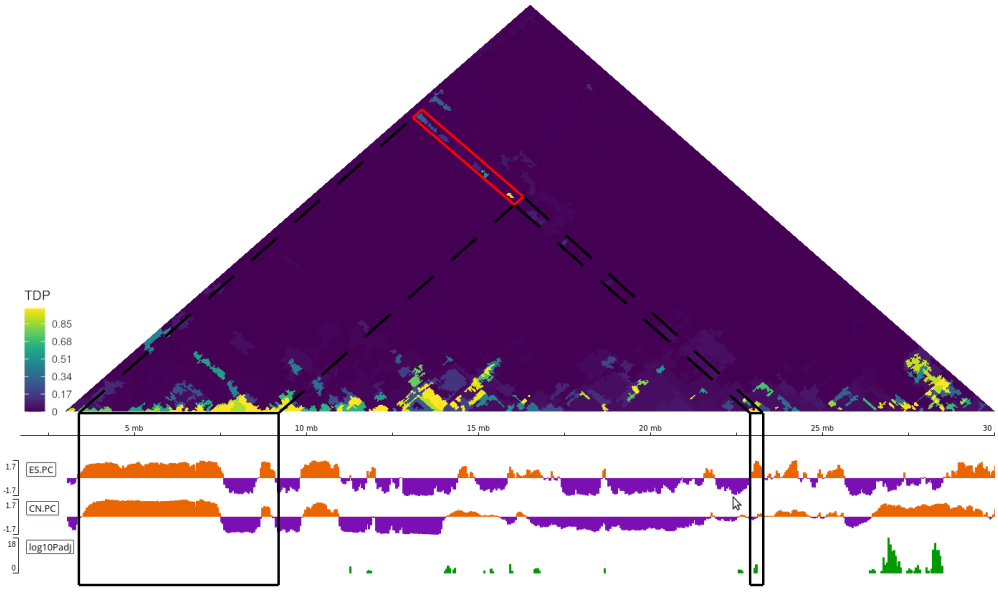


Figure S7: **Neuronal differentiation study: hicream and dcHiC results for specific section of chromosome 19.** (Top) **hicream** results for a specific part of chromosome 19 starting after 0Mb and ending at 30Mb.
(Bottom) **dcHiC** results for the same chromosome section, displaying compartment predicitons in ES and CN conditions as well as the (log10-adjusted) p-value for each bin. Orange values correspond to A compartment while purple values correspond to B compartment. A group of 7 selected clusters is outlined in red. The two specific parts of **dcHiC** results corresponding to the selected cluster projections are outlined in black.

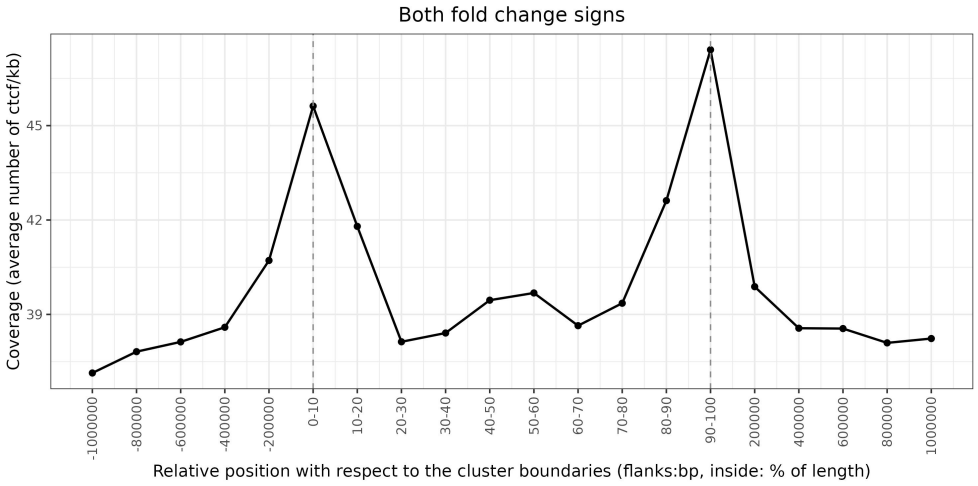


Figure S8: **Neuronal differentiation study: Average distribution of CTCF coverage in boundary regions of differential clusters across all chromosomes at 50kb resolution.** Positions on the x-axis correspond to left and right flanking regions of projected boundaries (each corresponding to five bins) and ten genomic subintervals of size 10% of the length between the two cluster boundaries.

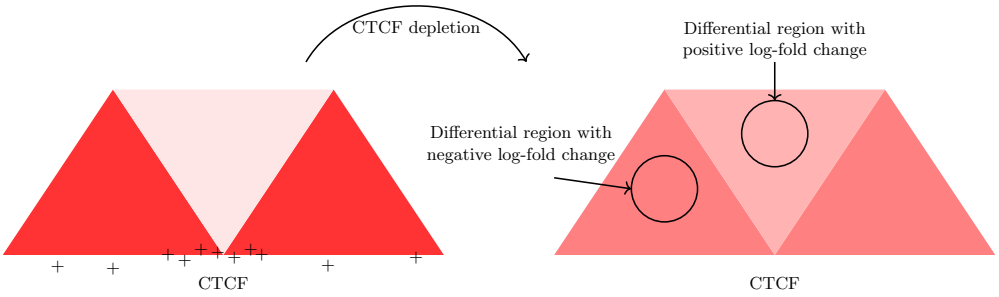


Figure S9: **Illustration of CTCF depletion effect.** On the left, two TAD-like regions are separated by a boundary. The CTCF density is high around that boundary but low within the two flanking regions. On the right, the induced CTCF depletion lowers the global density of CTCF and primarily impacts the boundary, allowing more frequent interactions between positions on each side of the boundary. This creates two types of differential regions with positive and negative log-fold change.

References

K. V. Ballman, D. E. Grill, A. L. Oberg, and T. M. Therneau. Faster cyclic loess: normalizing RNA arrays via linear models. *Bioinformatics*, 20(16):2778–2786, 2004. doi: 10.1093/bioinformatics/bth327.

B. Bonev, N. M. Cohen, Q. Szabo, L. Fritsch, G. L. Papadopoulos, Y. Lubling, X. Xu, X. Lv, J.-P. Hugnot, A. Tanay, and G. Cavalli. Multiscale 3D genome rewiring during mouse neural development. *Cell*, 171(3):557–572.e24, 2017. doi: 10.1016/j.cell.2017.09.043.

P. A. Ewels, A. Peltzer, S. Fillinger, H. Patel, J. Alneberg, A. Wilm, M. U. Garcia, P. Di Tommaso, and S. Nahnsen. The nf-core framework for community-curated bioinformatics pipelines. *Nature Biotechnology*, 38:276–278, 2020. doi: 10.1038/s41587-020-0439-x.

A. Gordon. A survey of constrained classification. *Computational Statistics & Data Analysis*, 21(1):17–29, 1996. doi: 10.1016/0167-9473(95)00005-4.

E. C. Grimm. CONISS: a FORTRAN 77 program for stratigraphically constrained analysis by the method of incremental sum of squares. *Computers & Geosciences*, 13(1):13–35, 1987. doi: 10.1016/0098-3004(87)90022-7.

L. Lebart. Programme d'agrégation avec contraintes. *Les Cahiers de l'Analyse des Données*, 3(3):275–287, 1978. URL `http://www.numdam.org/item?id=CAD_1978__3_3_275_0`.

A. T. Lun and G. K. Smyth. csaw: a Bioconductor package for differential binding analysis of ChIP-seq data using sliding windows. *Nucleic Acids Research*, 44(5):e45, 2016. doi: 10.1093/nar/gkv1191.

E. Maigné, C. Kurylo, M. Zytnicki, and S. Foissac. HiCDOC: chromatin compartment prediction and differential analysis from Hi-C data with replicates. bioRxiv preprint. DOI: 10.1101/2025.09.18.677058, 2025.

N. Randriamihamison, N. Vialaneix, and P. Neuvial. Applicability and interpretability of Ward's hierarchical agglomerative clustering with or without contiguity constraints. *Journal of Classification*, 38:363–389, 2021. doi: 10.1007/s00357-020-09377-y.

M. D. Robinson, D. J. McCarthy, and G. K. Smyth. edger: a bioconductor package for differential expression analysis of digital gene expression data. *bioinformatics*, 26(1):139–140, 2010. doi: 10.1093/bioinformatics/btp616.

The ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, 2012. doi: 10.1038/nature11247.

B. Thirion, G. Varoquaux, E. Dohmatob, and J.-B. Poline. Which fMRI clustering gives good brain parcellations? *Frontiers in Neuroscience*, 8:167, 2014. doi: 10.3389/fnins.2014.00167.

J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

H. Zhang, D. J. Emerson, T. G. Gilgenast, K. R. Titus, Y. Lan, P. Huang, D. Zhang, H. Wang, C. A. Keller, B. Giardine, R. C. Hardison, J. E. Phillips-Cremins, and G. A. Blobel. Chromatin structure dynamics during the mitosis-to-G1 phase transition. *Nature*, 576(7785):158–162, 2019. doi: 10.1038/s41586-019-1778-y.

H. Zhang, J. Lam, D. Zhang, Y. Lan, M. W. Vermunt, C. A. Keller, B. Giardine, R. C. Hardison, and G. A. Blobel. CTCF and transcription influence chromatin structure re-configuration after mitosis. *Nature Communications*, 12:5157, 2021. doi: 10.1038/s41467-021-25418-5.

Supplementary File S1

AME
Analysis of Motif Enrichment

For further information on how to interpret these results please access http://meme-suite.org/doc/ame-output-format.html.
To get a copy of the MEME software please access https://meme-suite.org.

If you use AME in your research, please cite the following paper:
Robert C. McLeay and Timothy L. Bailey, "Motif Enrichment Analysis: a unified framework and an evaluation on ChIP data", *BMC Bioinformatics*, **11**:165, 2010. [full text]

ENRICHED MOTIFS | INPUT FILES | PROGRAM INFORMATION | RESULTS IN TSV FORMAT | POSITIVE SEQUENCES FOR EACH MOTIF

ENRICHED MOTIFS

Sequence motif score: avg_odds
Background model source: letter frequencies in (primary) sequences file (res/ctcf/200000/clusterboundaries_mintpr0.1_200000_allfc_firstlastbins.fasta)
Background model frequencies: 0.285516,0.214484,0.214484,0.285516
Total pseudocount added to a motif column: 0.1

Statistical test: Fisher's exact test (optimized over motif scores)
Labeling positives: all 2406 primary sequences are labeled as 'positive'; all 2394 control sequences are labeled as 'negative'
Classifying positives: sequences with the motif scores ≥ 'TP Thresh' are classified as positive

Logo	Database	ID	Alt ID	*p*-value	*E*-value	Pos Thresh	Pos	Neg	TP Thresh	TP (%)	FP (%)	Pearsons's CC (ρ)	Spearman Rank CC (ρ)

No motifs passed the *E*-value threshold (10).

INPUT FILES

Alphabet

Background source: built from the (primary) sequences

Name	Bg.				Bg.	Name
Adenine	0.285516	A	~	T	0.285516	Thymine
Cytosine	0.214484	C	~	G	0.214484	Guanine

Sequences

Primary Sequences	Number	Control Sequences
res/ctcf/200000/clusterboundaries_mintpr0.1_200000_allfc_firstlastbins.fasta	2406	res/ctcf/200000/clusterboundaries_mintpr0.1_200000_allfc_fla

Motifs

Database	Source	Motif Count
JASPAR2024 CORE vertebrates non-redundant pfms meme	data/motif_databases/JASPAR/2024/JASPAR2024_CORE_vertebrates_non-redundant_pfms_meme.txt	879

AME version
5.5.4 (Release date: Fri Jun 16 12:19:08 2023 -0700)
Copyright © Robert McLeay & Timothy Bailey meme-suite@uw.edu, 2009.

Reference
Robert C. McLeay and Timothy L. Bailey, "Motif Enrichment Analysis: a unified framework and an evaluation on ChIP data", *BMC Bioinformatics*, **11**:165, 2010. [full text]

Command line

```
ame --oc res/ctcf/200000//flankingbackground_200000 --control
res/ctcf/200000/clusterboundaries_mintpr0.1_200000_allfc_flanks.fasta
res/ctcf/200000/clusterboundaries_mintpr0.1_200000_allfc_firstlastbins.fasta
data/motif_databases/JASPAR/2024/JASPAR2024_CORE_vertebrates_non-redundant_pfms_meme.txt
```

Supplementary File S2

AME
Analysis of Motif Enrichment

For further information on how to interpret these results please access http://meme-suite.org/doc/ame-output-format.html.
To get a copy of the MEME software please access https://meme-suite.org.

If you use AME in your research, please cite the following paper:
Robert C. McLeay and Timothy L. Bailey, "Motif Enrichment Analysis: a unified framework and an evaluation on ChIP data", *BMC Bioinformatics*, **11**:165, 2010. [full text]

ENRICHED MOTIFS | INPUT FILES | PROGRAM INFORMATION | RESULTS IN TSV FORMAT | POSITIVE SEQUENCES FOR EACH MOTIF

ENRICHED MOTIFS

Sequence motif score: avg_odds
Background model source: letter frequencies in (primary) sequences file (res/bonev/50000/clusterboundaries_mintpr0.1_50000_allfc_firstlastbins.fasta)
Background model frequencies: 0.284679,0.215321,0.215321,0.284679
Total pseudocount added to a motif column: 0.1

Statistical test: Fisher's exact test (optimized over motif scores)
Labeling positives: all 8502 primary sequences are labeled as 'positive'; all 8502 control sequences are labeled as 'negative'
Classifying positives: sequences with the motif scores ≥ 'TP Thresh' are classified as positive

E-value threshold for reporting results: 10

Logo	Database	ID	Alt ID	*p*-value	*E*-value	TP Thresh
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA0139.2	CTCF	8.87e-13	7.80e-10	4.43
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA1125.2	ZNF384	5.64e-7	4.96e-4	1.00
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA0747.2	SP8	3.08e-6	2.71e-3	1.10
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA1929.2	CTCF	6.99e-6	6.14e-3	29.06
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA2326.1	IKZF2	7.02e-6	6.17e-3	1.01
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA1723.2	PRDM9	7.75e-6	6.82e-3	1.04
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA1508.2	IKZF1	1.92e-5	1.69e-2	1.02
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA0136.4	Elf5	3.41e-5	2.99e-2	1.00
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA1623.2	Stat2	6.75e-5	5.93e-2	1.01
	JASPAR2024 CORE vertebrates	MA1116.2	RBPJ	1.12e-	9.81e-	1.00

	non-redundant pfms meme			4	2	
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA1965.2	SP5	1.89e-4	1.66e-1	1.17
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA1107.3	KLF9	4.60e-4	4.04e-1	1.03
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA0508.4	PRDM1	1.12e-3	9.88e-1	1.01
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA0002.3	Runx1	1.31e-3	1.15e0	1.00
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA0741.1	KLF16	2.07e-3	1.82e0	1.21
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA0080.7	Spi1	3.59e-3	3.16e0	1.01
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA0599.1	KLF5	4.22e-3	3.71e0	1.18
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA0493.3	KLF1	5.00e-3	4.39e0	1.04
	JASPAR2024 CORE vertebrates non-redundant pfms meme	MA1930.2	CTCF	6.28e-3	5.52e0	12.15

Input Files

Alphabet

Background source: built from the (primary) sequences

Name	Bg.				Bg.	Name
Adenine	0.284679	A	~	T	0.284679	Thymine
Cytosine	0.215321	C	~	G	0.215321	Guanine

Sequences

Primary Sequences	Number	Control Sequences
res/bonev/50000/clusterboundaries_mintpr0.1_50000_allfc_firstlastbins.fasta	8502	res/bonev/50000/clusterboundaries_mintpr0.1_50000_allfc_fla

Motifs

Database	Source	Motif Count
JASPAR2024 CORE vertebrates non-redundant pfms meme	data/motif_databases/JASPAR/2024/JASPAR2024_CORE_vertebrates_non-redundant_pfms_meme.txt	879

AME version

5.5.4 (Release date: Fri Jun 16 12:19:08 2023 -0700)
Copyright © Robert McLeay & Timothy Bailey meme-suite@uw.edu, 2009.

Reference

Robert C. McLeay and Timothy L. Bailey, "Motif Enrichment Analysis: a unified framework and an evaluation on ChIP data", *BMC Bioinformatics*, **11**:165, 2010. [full text]

Command line

```
ame --oc res/bonev/50000//flankingbackground_50000 --control
res/bonev/50000/clusterboundaries_mintpr0.1_50000_allfc_flanks.fasta
res/bonev/50000/clusterboundaries_mintpr0.1_50000_allfc_firstlastbins.fasta
data/motif_databases/JASPAR/2024/JASPAR2024_CORE_vertebrates_non-redundant_pfms_meme.txt
```

Supplementary Table S1

Analysis Type:	PANTHER Overrepresentation Test (Released 20240807)					
Annotation Version and Release Date:	PANTHER version 19.0 Released 2024-06-20					
Analyzed List:	upload_1 (Mus musculus)					
Reference List:	Mus musculus (all genes in database)					
Test Type:	FISHER					
Correction:	BONFERRONI					
Bonferroni count:		1772				
PANTHER GO-Slim Biological Process	Mus musculus - REFLIST (21836)	upload_1 (382)	upload_1 (expected)	upload_1 (over/under)	upload_1 (fold Enrichment)	upload_1 (P-value)
receptor signaling pathway via JAK-STAT (GO:0007259)		11	7	0.19 +	36.38	2.61E-07
receptor signaling pathway via STAT (GO:0097696)		11	7	0.19 +	36.38	2.61E-07
intracellular receptor signaling pathway (GO:0030522)		14	8	0.24 +	32.66	3.96E-08
calcineurin-mediated signaling (GO:0097720)		10	5	0.17 +	28.58	6.63E-04
cytokine production (GO:0001816)		10	5	0.17 +	28.58	6.63E-04
hormone-mediated signaling pathway (GO:0009755)		28	9	0.49 +	18.37	1.27E-06
anterior/posterior pattern specification (GO:0009952)		46	13	0.8 +	16.15	1.26E-09
sensory organ development (GO:0007423)		66	18	1.15 +	15.59	8.92E-14
regulation of neurogenesis (GO:0050767)		45	12	0.79 +	15.24	2.10E-08
regionalization (GO:0003002)		53	13	0.93 +	14.02	9.29E-09
regulation of cell development (GO:0060284)		68	15	1.19 +	12.61	1.13E-09
regulation of transcription by RNA polymerase II (GO:0006357)		1565	334	27.38 +	12.2	0.00E00
pattern specification process (GO:0007389)		66	14	1.15 +	12.13	1.19E-08
positive regulation of transcription by RNA polymerase II (GO:0045944)		229	45	4.01 +	11.23	8.06E-31
regulation of nervous system development (GO:0051960)		62	12	1.08 +	11.06	1.21E-06
regulation of DNA-templated transcription (GO:0006355)		1839	344	32.17 +	10.69	0.00E00
regulation of RNA biosynthetic process (GO:2001141)		1840	344	32.19 +	10.69	0.00E00
regulation of RNA metabolic process (GO:0051252)		1999	344	34.97 +	9.84	7.31E-316
regulation of nucleobase-containing compound metabolic process (GO:0019219)		2061	345	36.06 +	9.57	6.20E-313
positive regulation of DNA-templated transcription (GO:0045893)		279	45	4.88 +	9.22	6.52E-27
positive regulation of RNA biosynthetic process (GO:1902680)		279	45	4.88 +	9.22	6.52E-27
regulation of cell differentiation (GO:0045595)		112	18	1.96 +	9.19	1.78E-09
regulation of gene expression (GO:0010468)		2347	347	41.06 +	8.45	6.35E-296
regulation of macromolecule biosynthetic process (GO:0010556)		2370	347	41.46 +	8.37	2.37E-294
regulation of cellular biosynthetic process (GO:0031326)		2386	347	41.74 +	8.31	2.87E-293
regulation of biosynthetic process (GO:0009889)		2392	347	41.85 +	8.29	7.28E-293
response to peptide hormone (GO:0043434)		49	7	0.86 +	8.17	3.84E-02
negative regulation of transcription by RNA polymerase II (GO:0000122)		182	26	3.18 +	8.17	2.88E-13
regulation of nitrogen compound metabolic process (GO:0051171)		2449	345	42.84 +	8.05	3.39E-285
response to peptide (GO:1901652)		50	7	0.87 +	8	4.40E-02
regulation of primary metabolic process (GO:0080090)		2479	345	43.37 +	7.96	2.95E-283
positive regulation of RNA metabolic process (GO:0051254)		331	45	5.79 +	7.77	1.19E-23
cellular response to hormone stimulus (GO:0032870)		96	13	1.68 +	7.74	2.26E-05
response to hormone (GO:0009725)		121	16	2.12 +	7.56	6.72E-07
regulation of cellular metabolic process (GO:0031323)		2674	347	46.78 +	7.42	5.04E-275
positive regulation of nucleobase-containing compound metabolic process (GO:0045935)		357	46	6.25 +	7.37	3.28E-23
regulation of macromolecule metabolic process (GO:0060255)		2725	347	47.67 +	7.28	5.11E-272
head development (GO:0060322)		63	8	1.1 +	7.26	2.42E-02
brain development (GO:0007420)		63	8	1.1 +	7.26	2.42E-02
regulation of metabolic process (GO:0019222)		2870	347	50.21 +	6.91	8.58E-264
regulation of cytokine production (GO:0001817)		86	10	1.5 +	6.65	4.59E-03
cell differentiation (GO:0030154)		859	98	15.03 +	6.52	7.85E-49
cellular developmental process (GO:0048869)		859	98	15.03 +	6.52	7.85E-49
embryo development (GO:0009790)		88	10	1.54 +	6.5	5.68E-03

positive regulation of macromolecule biosynthetic process (GO:0010557)	402	45	7.03	+	6.4 4.45E-20
positive regulation of cellular biosynthetic process (GO:0031328)	409	45	7.16	+	6.29 9.08E-20
positive regulation of biosynthetic process (GO:0009891)	410	45	7.17	+	6.27 1.00E-19
negative regulation of DNA-templated transcription (GO:0045892)	237	26	4.15	+	6.27 1.81E-10
negative regulation of RNA biosynthetic process (GO:1902679)	238	26	4.16	+	6.24 2.00E-10
animal organ morphogenesis (GO:0009887)	102	11	1.78	+	6.16 3.09E-03
regulation of multicellular organismal development (GO:2000026)	142	15	2.48	+	6.04 5.41E-05
negative regulation of RNA metabolic process (GO:0051253)	257	26	4.5	+	5.78 1.21E-09
regulation of cell population proliferation (GO:0042127)	173	16	3.03	+	5.29 1.24E-04
developmental process (GO:0032502)	1411	130	24.68	+	5.27 2.98E-56
negative regulation of nucleobase-containing compound metabolic process (GO:0045934)	284	26	4.97	+	5.23 1.20E-08
axon development (GO:0061564)	151	13	2.64	+	4.92 4.72E-03
animal organ development (GO:0048513)	434	37	7.59	+	4.87 3.42E-12
positive regulation of nitrogen compound metabolic process (GO:0051173)	541	46	9.46	+	4.86 1.12E-15
regulation of multicellular organismal process (GO:0051239)	328	27	5.74	+	4.71 5.54E-08
positive regulation of cellular metabolic process (GO:0031325)	581	46	10.16	+	4.53 1.83E-14
neuron differentiation (GO:0030182)	306	24	5.35	+	4.48 1.89E-06
positive regulation of macromolecule metabolic process (GO:0010604)	589	46	10.3	+	4.46 3.12E-14
anatomical structure morphogenesis (GO:0009653)	436	34	7.63	+	4.46 6.52E-10
regulation of developmental process (GO:0050793)	251	19	4.39	+	4.33 1.86E-04
generation of neurons (GO:0048699)	319	24	5.58	+	4.3 4.33E-06
anatomical structure development (GO:0048856)	1252	89	21.9	+	4.06 1.68E-27
positive regulation of metabolic process (GO:0009893)	659	46	11.53	+	3.99 2.24E-12
neurogenesis (GO:0022008)	348	24	6.09	+	3.94 2.37E-05
neuron development (GO:0048666)	233	16	4.08	+	3.93 6.77E-03
response to endogenous stimulus (GO:0009719)	237	16	4.15	+	3.86 8.42E-03
regulation of cellular process (GO:0050794)	5475	351	95.78	+	3.66 3.52E-170
negative regulation of nitrogen compound metabolic process (GO:0051172)	425	26	7.43	+	3.5 6.76E-05
multicellular organism development (GO:0007275)	819	50	14.33	+	3.49 2.38E-11
regulation of biological process (GO:0050789)	5834	353	102.06	+	3.46 1.54E-163
negative regulation of macromolecule biosynthetic process (GO:0010558)	434	26	7.59	+	3.42 1.03E-04
negative regulation of biosynthetic process (GO:0009890)	438	26	7.66	+	3.39 1.23E-04
negative regulation of cellular biosynthetic process (GO:0031327)	438	26	7.66	+	3.39 1.23E-04
system development (GO:0048731)	675	39	11.81	+	3.3 1.24E-07
biological regulation (GO:0065007)	6133	354	107.29	+	3.3 2.13E-157
nervous system development (GO:0007399)	468	27	8.19	+	3.3 1.23E-04
negative regulation of cellular metabolic process (GO:0031324)	537	26	9.39	+	2.77 5.86E-03
negative regulation of macromolecule metabolic process (GO:0010605)	587	26	10.27	+	2.53 2.82E-02
positive regulation of cellular process (GO:0048522)	1180	51	20.64	+	2.47 5.05E-06
negative regulation of metabolic process (GO:0009892)	606	26	10.6	+	2.45 4.85E-02
positive regulation of biological process (GO:0048518)	1383	52	24.19	+	2.15 3.41E-04
multicellular organismal process (GO:0032501)	1760	58	30.79	+	1.88 4.58E-03
biological_process (GO:0008150)	13046	361	228.23	+	1.58 8.82E-53
cellular component organization (GO:0016043)	2220	13	38.84	-	0.33 1.51E-03
cellular component organization or biogenesis (GO:0071840)	2337	13	40.88	-	0.32 2.91E-04
cellular biosynthetic process (GO:0044249)	1734	7	30.33	-	0.23 4.92E-04
organic substance biosynthetic process (GO:1901576)	1778	7	31.1	-	0.23 2.42E-04
biosynthetic process (GO:0009058)	1782	7	31.17	-	0.22 2.42E-04
cellular metabolic process (GO:0044237)	3153	8	55.16	-	0.15 1.92E-13
macromolecule metabolic process (GO:0043170)	2903	7	50.79	-	0.14 2.69E-12
Unclassified (UNCLASSIFIED)	8790	21	153.77	-	0.14 0.00E00
nucleic acid metabolic process (GO:0090304)	902	2	15.78	-	0.13 4.19E-02
organic substance metabolic process (GO:0071704)	3947	8	69.05	-	0.12 1.08E-19
metabolic process (GO:0008152)	4181	8	73.14	-	0.11 8.90E-22
nucleobase-containing compound metabolic process (GO:0006139)	1130	2	19.77	-	0.1 9.81E-04
heterocycle metabolic process (GO:0046483)	1187	2	20.77	-	0.1 4.76E-04
cellular aromatic compound metabolic process (GO:0006725)	1195	2	20.91	-	0.1 3.10E-04

organic cyclic compound metabolic process (GO:1901360)	1284	2	22.46	-	0.09 6.59E-05
cellular nitrogen compound metabolic process (GO:0034641)	1349	2	23.6	-	0.08 3.24E-05
transport (GO:0006810)	1694	2	29.63	-	0.07 5.00E-08
establishment of localization (GO:0051234)	1792	2	31.35	-	0.06 9.78E-09
localization (GO:0051179)	1981	2	34.66	-	0.06 3.59E-10
primary metabolic process (GO:0044238)	3509	3	61.39	-	0.05 8.79E-22
nitrogen compound metabolic process (GO:0006807)	3166	2	55.39	-	0.04 4.27E-20

Bibliographie

Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell*. New York, 4 edition, 2002. ISBN 0-8153-3218-1, 0-8153-4072-9. URL https://www.ncbi.nlm.nih.gov/books/NBK26821/. Available from NCBI Bookshelf.

Christophe Ambroise, Alia Dehman, Pierre Neuvial, Guillem Rigaill, and Nathalie Vialaneix. Adjacency-constrained hierarchical clustering of a band similarity matrix with application to genomics. *Algorithms for Molecular Biology*, 14(1) :22, November 2019. doi : 10.1186/s13015-019-0157-4.

Abbas Roayaei Ardakany, Ferhat Ay, and Stefano Lonardi. Selfish : discovery of differential chromatin interactions via a self-similarity measure. *Bioinformatics*, 35(14) :i145–i153, 2019. doi : 10.1093/bioinformatics/btz362.

Jon-Matthew Belton, Rachel Patton McCord, Johan Harmen Gibcus, Natalia Naumova, Ye Zhan, and Job Dekker. Hi–C : A comprehensive technique to capture the conformation of genomes. *Methods*, 58(3) :268–276, November 2012. doi : 10.1016/j.ymeth.2012.05.001.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate : A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1) :289–300, 1995.

Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, 29(4) :1165–1188, 2001. doi : 10.1214/aos/1013699998.

Alexandre Blain, Bertrand Thirion, and Pierre Neuvial. Notip : Non-parametric true discovery proportion control for brain imaging. *NeuroImage*, 260 :119492, 2022. doi : 10.1016/j.neuroimage.2022.119492.

Gilles Blanchard, Pierre Neuvial, and Etienne Roquain. Post hoc confidence bounds on false positives using reference families. *The Annals of Statistics*, 48(3) :1281–1303, 2020. doi : 10.1214/19-AOS1847.

Boyan Bonev, Netta Mendelson Cohen, Quentin Szabo, Lauriane Fritsch, Giorgio L Papadopoulos, Yaniv Lubling, Xiaole Xu, Xiaodan Lv, Jean-Philippe Hugnot, Amos Tanay, and Giacomo Cavalli. Multiscale 3D genome rewiring during mouse neural development. *Cell*, 171(3) :557–572.e24, 2017. doi : 10.1016/j.cell.2017.09.043.

Carlo Bonferroni. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8 :3–62, 1936.

Richard Bourgon, Robert Gentleman, and Wolfgang Huber. Independent filtering increases detection power for high-throughput experiments. *Proceedings of the national academy of sciences*, 107(21) :9546–9551, 2010.

Abhijit Chakraborty, Jeffrey G. Wang, and Ferhat Ay. dcHiC detects differential compartments across multiple Hi-C datasets. *Nature Communications*, 13(1) :6827, November 2022. doi : 10.1038/s41467-022-34626-6.

Fengling Chen, Guipeng Li, Michael Q Zhang, and Yang Chen. HiCDB : a sensitive and robust method for detecting contact domain boundaries. *Nucleic Acids Research*, 46 (21) :11239–11250, November 2018. ISSN 0305-1048. doi : 10.1093/nar/gky789.

Yunshun Chen, Lizhong Chen, Aaron T L Lun, Pedro L Baldoni, and Gordon K Smyth. edgeR v4 : powerful differential analysis of sequencing data with expanded functionality and improved support for small counts and larger datasets. *Nucleic Acids Research*, 53 (2) :gkaf018, January 2025. doi : 10.1093/nar/gkaf018.

H. M. A. Mohit Chowdhury, Terrance Boult, and Oluwatosin Oluwadare. Comparative study on chromatin loop callers using Hi-C data reveals their effectiveness. *BMC Bioinformatics*, 25(1) :123, March 2024. doi : 10.1186/s12859-024-05713-w.

William S. Cleveland. Robust Locally Weighted Regression and Smoothing Scatterplots. *Journal of the American Statistical Association*, 74(368) :829–836, 1979. ISSN 0162-1459. doi : 10.2307/2286407.

Axel Cournac, Hervé Marie-Nelly, Martial Marbouty, Romain Koszul, and Julien Mozziconacci. Normalization of a chromosomal contact map. *BMC Genomics*, 13(1) :436, August 2012. doi : 10.1186/1471-2164-13-436.

T. Cremer and C. Cremer. Chromosome territories, nuclear architecture and gene regulation in mammalian cells. *Nature Reviews Genetics*, 2(4) :292–301, April 2001. doi : 10.1038/35066075.

T. Cremer, A. Kurz, R. Zirbel, S. Dietzel, B. Rinke, E. Schröck, M. R. Speicher, U. Mathieu, A. Jauch, P. Emmerich, H. Scherthan, T. Ried, C. Cremer, and P. Lichter. Role of chromosome territories in the functional compartmentalization of the cell nucleus. *Cold Spring Harbor Symposia on Quantitative Biology*, 58 :777–792, 1993. doi : 10.1101/sqb.1993.058.01.085.

Kellen G. Cresswell and Mikhail G. Dozmorov. TADCompare : An R Package for Differential and Temporal Analysis of Topologically Associated Domains. *Frontiers in Genetics*, 11, March 2020. doi : 10.3389/fgene.2020.00158.

Rola Dali and Mathieu Blanchette. A critical assessment of topologically associating domain prediction tools. *Nucleic Acids Research*, 45(6) :2994–3005, April 2017. doi : 10.1093/nar/gkx145.

Priyojit Das, Tongye Shen, and Rachel Patton McCord. Inferring chromosome radial organization from Hi-C data. *BMC Bioinformatics*, 21(1) :511, November 2020. doi : 10.1186/s12859-020-03841-7.

Job Dekker, Karsten Rippe, Martijn Dekker, and Nancy Kleckner. Capturing chromosome conformation. *Science (New York, N.Y.)*, 295(5558) :1306–1311, February 2002. doi : 10.1126/science.1067799.

Jesse R. Dixon, Siddarth Selvaraj, Feng Yue, Audrey Kim, Yan Li, Yin Shen, Ming Hu, Jun S. Liu, and Bing Ren. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, 485(7398) :376–380, May 2012. doi : 10.1038/nature11082.

Sandrine Dudoit, Yee Hwa Yang, Matthew J. Callow, and Terence P. Speed. STATISTICAL METHODS FOR IDENTIFYING DIFFERENTIALLY EXPRESSED GENES IN REPLICATED cDNA MICROARRAY EXPERIMENTS. *Statistica Sinica*, 12(1) : 111–139, 2002. URL https://www.jstor.org/stable/24307038.

Guillermo Durand, Gilles Blanchard, Pierre Neuvial, and Etienne Roquain. Post hoc false positive control for structured hypotheses. *Scandinavian Journal of Statistics*, 47(4) : 1114–1148, December 2020. doi : 10.1111/sjos.12453.

Neva C. Durand, Muhammad S. Shamim, Ido Machol, Suhas S. P. Rao, Miriam H. Huntley, Eric S. Lander, and Erez Lieberman Aiden. Juicer Provides a One-Click System for Analyzing Loop-Resolution Hi-C Experiments. *Cell Systems*, 3(1) :95–98, July 2016. doi : 10.1016/j.cels.2016.07.002.

Mitra Ebrahimpoor and Jelle J Goeman. Inflated false discovery rate due to volcano plots : problem and solutions. *Briefings in Bioinformatics*, 22(5) :bbab053, 2021.

Nicolas Enjalbert-Courrech and Pierre Neuvial. Powerful and interpretable control of false discoveries in differential expression studies. *Bioinformatics*, 38(23) :5214–5221, 2022. doi : 10.1093/bioinformatics/btac693.

Mattia Forcato, Chiara Nicoletti, Koustav Pal, Carmen Maria Livi, Francesco Ferrari, and Silvio Bicciato. Comparison of computational methods for Hi-C data analysis. *Nature Methods*, 14(7) :679–685, July 2017. doi : 10.1038/nmeth.4325.

Geoff Fudenberg, David R. Kelley, and Katherine S. Pollard. Predicting 3D genome folding from DNA sequence with Akita. *Nature Methods*, 17(11) :1111–1117, November 2020. doi : 10.1038/s41592-020-0958-x.

Geoffrey Fudenberg, Maxim Imakaev, Carolyn Lu, Anton Goloborodko, Nezar Abdennur, and Leonid A. Mirny. Formation of Chromosomal Domains by Loop Extrusion. *Cell Reports*, 15(9) :2038–2049, May 2016. doi : 10.1016/j.celrep.2016.04.085.

Xinzhou Ge, Yiling Elaine Chen, Dongyuan Song, MeiLu McDermott, Kyla Woyshner, Antigoni Manousopoulou, Ning Wang, Wei Li, Leo D. Wang, and Jingyi Jessica Li. Clipper : p-value-free FDR control on high-throughput data from two conditions. *Genome Biology*, 22 :288, October 2021. doi : 10.1186/s13059-021-02506-9.

Jelle J. Goeman and Aldo Solari. Multiple testing for exploratory research. *Statistical Science*, 26(4) :584–597, 2011. doi : 10.1214/11-STS356.

Jelle J. Goeman and Aldo Solari. Multiple hypothesis testing in genomics. *Statistics in Medicine*, 33(11) :1946–1978, 2014. doi : 10.1002/sim.6082.

Haiyan Gong, Dawei Zhang, and Xiaotong Zhang. TOAST : A novel method for identifying topologically associated domains based on graph auto-encoders and clustering. *Computational and Structural Biotechnology Journal*, 21 :4759–4768, January 2023. doi : 10.1016/j.csbj.2023.09.019.

Hélène Hayes. ADN et chromosomes. *INRAE Productions Animales*, 13(HS) :13–20, 2000. doi : 10.20870/productions-animales.2000.13.HS.3806.

Erica M. Hildebrand and Job Dekker. Mechanisms and Functions of Chromosome Compartmentalization. *Trends in Biochemical Sciences*, 45(5) :385–396, May 2020. doi : 10.1016/j.tibs.2020.01.002.

Yosef Hochberg. A sharper Bonferroni procedure for multiple tests of significance. *Biometrika*, 75(4) :800–802, December 1988. doi : 10.1093/biomet/75.4.800.

Sture Holm. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*, 6(2) :65–70, 1979.

Gerhard Hommel. A stagewise rejective multiple test procedure based on a modified bonferroni test. *Biometrika*, pages 383–386, 1988.

Ming Hu, Ke Deng, Siddarth Selvaraj, Zhaohui Qin, Bing Ren, and Jun S. Liu. HiCNorm : removing biases in Hi-C data via Poisson regression. *Bioinformatics*, 28(23) :3131–3133, December 2012. doi : 10.1093/bioinformatics/bts570.

Dunming Hua, Ming Gu, Xiao Zhang, Yanyi Du, Hangcheng Xie, Li Qi, Xiangjun Du, Zhidong Bai, Xiaopeng Zhu, and Dechao Tian. DiffDomain enables identification of structurally reorganized topologically associating domains. *Nature Communications*, 15(1) :502, January 2024. doi : 10.1038/s41467-024-44782-6.

Maxim Imakaev, Geoffrey Fudenberg, Rachel Patton McCord, Natalia Naumova, Anton Goloborodko, Bryan R. Lajoie, Job Dekker, and Leonid A. Mirny. Iterative correction of Hi-C data reveals hallmarks of chromosome organization. *Nature Methods*, 9(10) : 999–1003, October 2012. doi : 10.1038/nmeth.2148.

Ivana Jerković and Giacomo Cavalli. Understanding 3D genome organization by multidisciplinary methods. *Nature Reviews Molecular Cell Biology*, 22(8) :511–528, August 2021. doi : 10.1038/s41580-021-00362-w.

Elise Jorge, Sylvain Foissac, Pierre Neuvial, Matthias Zytnicki, and Nathalie Vialaneix. A comprehensive review and benchmark of differential analysis tools for Hi-C data. *Briefings in Bioinformatics*, 26(2) :bbaf074, 2025. doi : 10.1093/bib/bbaf074.

Elise Jorge, Toby Dylan Hocking, Pierre Neuvial, Nathalie Vialaneix, and Sylvain Foissac. hicream : a flexible framework to identify significantly different regions in hi-c data. *Bioinformatics*, August 2026. doi : 10.1093/bioinformatics/btag631.

Yan Kai, Nan Liu, Stuart H. Orkin, and Guo-Cheng Yuan. Identifying quantitatively differential chromosomal compartmentalization changes and their biological significance from Hi-C data using DARIC. *BMC Genomics*, 24(1) :614, October 2023. doi : 10.1186/s12864-023-09675-w.

Gopal K. Kanji. *100 Statistical Tests*. SAGE Publications Ltd, London, 3 edition, 2006. ISBN 9781412923767. doi : 10.4135/9781849208499.

Rieke Kempfer and Ana Pombo. Methods for mapping 3D chromosome architecture. *Nature Reviews Genetics*, 21(4) :207–226, April 2020. doi : 10.1038/s41576-019-0195-2.

Philip A. Knight and Daniel Ruiz. A fast algorithm for matrix balancing. *IMA Journal of Numerical Analysis*, 33(3) :1029–1047, July 2013. doi : 10.1093/imanum/drs019.

Vipin Kumar, Simon Leclerc, and Yuichi Taniguchi. BHi-Cect : a top-down algorithm for identifying the multi-scale hierarchical structure of chromosomes. *Nucleic Acids Research*, 48(5) :e26, March 2020. doi : 10.1093/nar/gkaa004.

Bryan R. Lajoie, Job Dekker, and Noam Kaplan. The Hitchhiker's guide to Hi-C analysis : Practical guidelines. *Methods*, 72 :65–75, January 2015. doi : 10.1016/j.ymeth.2014.10.031.

Erez Lieberman-Aiden, Nynke L. van Berkum, Louise Williams, Maxim Imakaev, Tobias Ragoczy, Agnes Telling, Ido Amit, Bryan R. Lajoie, Peter J. Sabo, Michael O. Dorschner, Richard Sandstrom, Bradley Bernstein, M. A. Bender, Mark Groudine, Andreas Gnirke, John Stamatoyannopoulos, Leonid A. Mirny, Eric S. Lander, and Job Dekker. Comprehensive Mapping of Long-Range Interactions Reveals Folding Principles of the Human Genome. *Science*, 326(5950) :289–293, October 2009. doi : 10.1126/science.1181369.

Huiling Liu and Wenxiu Ma. DiffGR : Detecting Differentially Interacting Genomic Regions from Hi-C Contact Maps. *Genomics, Proteomics & Bioinformatics*, 22(2) : qzae028, April 2024. doi : 10.1093/gpbjnl/qzae028.

Li Liu, Kaiyuan Han, Huimin Sun, Lu Han, Dong Gao, Qilemuge Xi, Lirong Zhang, and Hao Lin. A comprehensive review of bioinformatics tools for chromatin loop calling. *Briefings in Bioinformatics*, 24(2) :bbad072, March 2023. doi : 10.1093/bib/bbad072.

Michael I. Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12) :550, December 2014. doi : 10.1186/s13059-014-0550-8.

Aaron T.L. Lun and Gordon K. Smyth. diffHic : a Bioconductor package to detect differential genomic interactions in Hi-C data. *BMC Bioinformatics*, 16 :258, 2015. doi : 10.1186/s12859-015-0683-0.

Darío G. Lupiáñez, Katerina Kraft, Verena Heinrich, Peter Krawitz, Francesco Brancati, Eva Klopocki, Denise Horn, Hülya Kayserili, John M. Opitz, Renata Laxova, Fernando Santos-Simarro, Brigitte Gilbert-Dussardier, Lars Wittler, Marina Borschiwer, Stefan A. Haas, Marco Osterwalder, Martin Franke, Bernd Timmermann, Jochen Hecht, Malte Spielmann, Axel Visel, and Stefan Mundlos. Disruptions of topological chromatin domains cause pathogenic rewiring of gene-enhancer interactions. *Cell*, 161 (5) :1012–1025, 2015. doi : 10.1016/j.cell.2015.04.004.

Hongqiang Lyu, Lin Li, Zhifang Wu, Tian Wang, Jiguang Zheng, and Hongda Wang. TADBD : A Sensitive and Fast Method for Detection of Typologically Associated Domain Boundaries. *BioTechniques*, 69(1) :18–25, July 2020. doi : 10.2144/btn-2019-0165. _eprint : https://doi.org/10.2144/btn-2019-0165.

Celine Lévy-Leduc, M. Delattre, T. Mary-Huard, and S. Robin. Two-dimensional segmentation for analyzing Hi-C data. *Bioinformatics*, 30(17) :i386–i392, September 2014. ISSN 1367-4803. doi : 10.1093/bioinformatics/btu443.

Elise Maigné, Cyril Kurylo, Matthias Zytnicki, and Sylvain Foissac. HiCDOC : chromatin compartment prediction and differential analysis from Hi-C data with replicates, September 2025. ISSN : 2692-8205 Pages : 2025.09.18.677058 Section : New Results.

Karen J. Meaburn and Tom Misteli. Chromosome territories. *Nature*, 445(7126) :379–381, January 2007. doi : 10.1038/445379a.

Raphaël Mourad. TADreg : a versatile regression framework for TAD identification, differential analysis and rearranged 3D genome prediction. *BMC Bioinformatics*, 23 (1) :82, March 2022. doi : 10.1186/s12859-022-04614-0.

Kota Nagasaka, Iain F. Davidson, Roman R. Stocsits, Wen Tang, Gordana Wutz, Paul Batty, Melanie Panarotto, Gabriele Litos, Alexander Schleiffer, Daniel W. Gerlich, and Jan-Michael Peters. Cohesin mediates DNA loop extrusion and sister chromatid cohesion by distinct mechanisms. *Molecular Cell*, 83(17) :3049–3063.e6, September 2023. doi : 10.1016/j.molcel.2023.07.024.

Koustav Pal, Mattia Forcato, and Francesco Ferrari. Hi-C analysis : from data generation to integration. *Biophysical Reviews*, 11(1) :67–78, February 2019. ISSN 1867-2450. doi : 10.1007/s12551-018-0489-1.

Petter Portin and Adam Wilkins. The Evolving Definition of the Term “Gene”. *Genetics*, 205(4) :1353–1364, April 2017. doi : 10.1534/genetics.116.196956.

Suhas S. P. Rao, Miriam H. Huntley, Neva C. Durand, Elena K. Stamenova, Ivan D. Bochkov, James T. Robinson, Adrian L. Sanborn, Ido Machol, Arina D. Omer, Eric S. Lander, and Erez Lieberman Aiden. A 3D Map of the Human Genome at Kilobase Resolution Reveals Principles of Chromatin Looping. *Cell*, 159(7) :1665–1680, December 2014. doi : 10.1016/j.cell.2014.11.021.

Suhas S. P. Rao, Su-Chen Huang, Brian Glenn St Hilaire, Jesse M. Engreitz, Elizabeth M. Perez, Kyong-Rim Kieffer-Kwon, Adrian L. Sanborn, Sarah E. Johnstone, Gavin D. Bascom, Ivan D. Bochkov, Xingfan Huang, Muhammad S. Shamim, Jaeweon Shin, Douglass Turner, Ziyi Ye, Arina D. Omer, James T. Robinson, Tamar Schlick, Bradley E. Bernstein, Rafael Casellas, Eric S. Lander, and Erez Lieberman Aiden. Cohesin Loss Eliminates All Loop Domains. *Cell*, 171(2) :305–320.e24, October 2017. doi : 10.1016/j.cell.2017.09.026.

Ron Schwessinger, Matthew Gosden, Damien Downes, Richard C. Brown, A. Marieke Oudelaar, Jelena Telenius, Yee Whye Teh, Gerton Lunter, and Jim R. Hughes. DeepC : predicting 3D genome folding using megabase-scale transfer learning. *Nature Methods*, 17(11) :1118–1124, November 2020. doi : 10.1038/s41592-020-0960-3.

Emre Sefer. A comparison of topologically associating domain callers over mammals at high resolution. *BMC Bioinformatics*, 23 :127, April 2022. doi : 10.1186/s12859-022-04674-2.

Natalia Sikorska and Tom Sexton. Defining Functionally Relevant Spatial Chromatin Domains : It is a TAD Complicated. *Journal of Molecular Biology*, 432(3) :653–664, February 2020. doi : 10.1016/j.jmb.2019.12.006.

R. John Simes. An improved Bonferroni procedure for multiple tests of significance. *Biometrika*, 73(3) :751–754, 1986.

Paula Soler-Vila, Pol Cuscó, Irene Farabella, Marco Di Stefano, and Marc A Marti-Renom. Hierarchical chromatin organization detected by TADpole. *Nucleic Acids Research*, 48(7) :e39, April 2020. doi : 10.1093/nar/gkaa087.

John C. Stansfield, Kellen G. Cresswell, Vladimir I. Vladimirov, and Mikhail G. Dozmorov. HiCcompare : an R-package for joint normalization and comparison of HI-C datasets. *BMC Bioinformatics*, 19(1) :279, July 2018. doi : 10.1186/s12859-018-2288-x.

John C. Stansfield, Kellen G. Cresswell, and Mikhail G. Dozmorov. multiHiCcompare : joint normalization and comparative analysis of complex Hi-C experiments. *Bioinformatics*, 2019. doi : 10.1093/bioinformatics/btz048.

Philippe Tassi. *Méthodes Statistiques*. Economica, 3 edition, 2004. ISBN 9782717848595.

Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. Constrained K-means Clustering with Background Knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 577–584, San Francisco, CA, USA, June 2001. Morgan Kaufmann Publishers Inc. ISBN 978-1-55860-778-1.

Lukas M Weber, Wouter Saelens, Robrecht Cannoodt, Charlotte Soneson, Alexander Hapfelmeier, Paul P Gardner, Anne-Laure Boulesteix, Yvan Saeys, and Mark D Robinson. Essential guidelines for computational method benchmarking. *Genome biology*, 20(1) : 125, 2019.

Jingxuan Xu, Xiang Xu, Dandan Huang, Yawen Luo, Lin Lin, Xuemei Bai, Yang Zheng, Qian Yang, Yu Cheng, An Huang, Jingyi Shi, Xiaochen Bo, Jin Gu, and Hebing Chen.

A comprehensive benchmarking with interpretation and operational guidance for the hierarchy of topologically associating domains. *Nature Communications*, 15(1) :4376, May 2024. doi : 10.1038/s41467-024-48593-7.

Eitan Yaffe and Amos Tanay. Probabilistic modeling of Hi-C contact maps eliminates systematic biases to characterize global chromosomal architecture. *Nature Genetics*, 43 (11) :1059–1065, November 2011. doi : 10.1038/ng.947.

Rui Yang, Arnav Das, Vianne R. Gao, Alireza Karbalayghareh, William S. Noble, Jeffrey A. Bilmes, and Christina S. Leslie. Epiphany : predicting Hi-C contact maps from 1D epigenomic signals. *Genome Biology*, 24(1) :134, June 2023. doi : 10.1186/s13059-023-02934-9.

Daniel Yekutieli and Yoav Benjamini. Resampling-based false discovery rate controlling multiple test procedures for correlated test statistics. *Journal of Statistical Planning and Inference*, 82(1) :171–196, December 1999. doi : 10.1016/S0378-3758(99)00041-5.

Shilu Zhang, Deborah Chasman, Sara Knaack, and Sushmita Roy. In silico prediction of high-resolution Hi-C interaction matrices. *Nature Communications*, 10(1) :5449, December 2019. doi : 10.1038/s41467-019-13423-8.

Ye Zheng and Sündüz Keleş. FreeHi-C simulates high-fidelity Hi-C data for benchmarking and data augmentation. *Nature Methods*, 17(1) :37–40, January 2020. doi : 10.1038/s41592-019-0624-3.

Ye Zheng, Peigen Zhou, and Sündüz Keleş. FreeHi-C spike-in simulations for benchmarking differential chromatin interaction detection. *Methods*, 189 :3–11, May 2021. doi : 10.1016/j.ymeth.2020.07.001.

Marie Zufferey, Daniele Tavernari, Elisa Oricchio, and Giovanni Ciriello. Comparison of computational methods for the identification of topologically associating domains. *Genome Biology*, 19(1) :217, December 2018. doi : 10.1186/s13059-018-1596-9.

RÉPUBLIQUE FRANÇAISE
Liberté Égalité Fraternité

Université de Toulouse

Titre : Analyse différentielle de données Hi-C : évaluation d'outils et développement d'une méthode statistique reposant sur l'inférence post hoc

Mots clés : Génomique 3D, Hi-C, Analyse différentielle, Inférence post hoc, Tests statistiques

Résumé : De par la contrainte spatiale imposée par la taille du noyau dans lequel il est contenu, le génome se retrouve replié en trois dimensions (3D). Ce repliement induit la formation de structures organisées de manière hiérarchique. De ce fait, il est nécessaire d'aller au-delà d'une vision linéaire du génome pour comprendre le lien entre structures tridimensionnelles et fonctions associées. La technique Hi-C permet de cartographier l'organisation 3D du génome. L'analyse différentielle des données Hi-C, c'est-à-dire la comparaison des données issues de cette expérience entre plusieurs conditions biologiques, peut mettre en évidence des réarrangements dans la conformation spatiale du génome. Les résultats de ce type d'analyse contribuent à comprendre l'impact des réorganisations spatiales sur des processus tels que la régulation de l'expression des gènes ou la différenciation cellulaire.
Dans une première partie de la thèse, nous présentons un travail de comparaison de méthodes d'analyse différentielle de données Hi-C basées sur les pixels. Ce type de méthode effectue, pour chaque paire de positions génomiques, un test statistique mesurant la différence d'interaction entre les deux conditions biologiques étudiées. Nous avons réalisé une typologie des outils en fonction d'aspects techniques (implémentation, format de données en entrée, etc.) ainsi que méthodologiques (filtrages, normalisations, type de modèle statistique utilisé, type de prise en compte de la multiplicité des tests, etc.). Ensuite, nous nous sommes intéressés à l'évaluation des différents outils sur un jeu de données ne contenant pas de pixels différentiels, permettant de vérifier le contrôle correct de l'erreur de type I. Puis, nous avons évalué ces outils sur des données contenant des pixels différentiels et dont la position est connue, afin d'évaluer l'erreur de type II et donc la puissance de détection des outils. Cette comparaison a permis de mettre en avant les performances supérieures de la méthode diffHiC sur le contrôle de l'erreur de type I et la puissance de détection, par rapport aux autres outils.
Dans une seconde partie, nous présentons une nouvelle approche d'analyse différentielle – hicream – visant à identifier des régions différentielles de forme arbitraire. Cette approche a été motivée par les limitations des méthodes existantes d'analyse différentielle de données Hi-C. D'une part, les approches d'analyse différentielle basées sur les structures ne permettent pas de découvrir de nouvelles régions différentielles et ne fournissent pas toujours de garanties statistiques. D'autre part, les méthodes d'analyse basées sur les pixels produisent des résultats souvent peu interprétables avec des détections différentielles ponctuelles qui ne peuvent être généralisées à l'identification de régions différentielles. La méthode hicream est entièrement guidée par les données car elle définit les régions candidates au moyen d'un clustering. Ensuite, grâce aux résultats des tests fournis par diffHiC, elle s'appuie sur l'inférence post hoc pour évaluer chaque région candidate et fournir une proportion minimale de pixels différentiels dans chacune d'entre elles, permettant ainsi d'identifier les plus différentielles. La pertinence de la méthode a été démontrée à la fois sur des données semi-simulées et sur des données réelles en lien avec des signaux biologiques extérieurs. D'autre part, en comparant nos résultats à des outils d'identification de structures différentielles, nous avons pu montrer que nous pouvions trouver des clusters différentiels qui correspondent à des structures connues dans les données Hi-C telles que des TADs (Topologically Associating Domains) ou des compartiments. Enfin, l'interprétation des clusters différentiels détectés, de formes arbitraires, est facilitée par le développement d'une visualisation interactive des résultats.

Title: Differential analysis of Hi-C data : evaluation of tools and development of a statistical method based on post hoc inference

Key words: 3D genomics, Hi-C, Differential analysis, Post hoc inference, Statistical testing

Abstract: Due to spatial constraints imposed by the size of the nucleus in which it is contained, the genome is folded in three dimensions (3D). This folding leads to the formation of hierarchically organized structures. As a result, it is necessary to move beyond a linear view of the genome to understand the relationship between three-dimensional structures and their associated functions. The Hi-C technique allows for the mapping of the genome's 3D organization. Differential analysis of Hi-C data – that is, comparing data from the experiment across multiple biological conditions – can reveal rearrangements in the genome's spatial conformation. The results of this analysis contribute to understanding the impact of spatial reorganizations on biological processes such as gene expression regulation or cell differentiation.
In the first part of my thesis, we present a benchmark study of pixel-based differential analysis methods for Hi-C data. This type of method performs, for each pair of genomic positions, a statistical test measuring the difference in interaction between the two biological conditions under study. We provided a typology of the tools in terms of their technical aspects (implementation, input data format, etc.) as well as their methodological aspects (filtering, normalization, type of statistical model used, type of multiple testing correction, etc.). Next, we evaluated the various tools on a dataset with no differential pixels to assess type I error control. Then, we evaluated the tools on a dataset containing differential pixels with known positions in order to assess type II error and thus the tools' detection power. This benchmark highlighted the superior performance of the diffHiC method in type I error control and in detection power, compared with the other tools.
In the second part, we present a new differential analysis approach – hicream – aimed at identifying differential regions of arbitrary shapes. This approach was motivated by the limitations of existing Hi-C data differential analysis methods. On the one hand, structure-based differential analysis approaches do not allow for the discovery of new differential regions and do not always provide statistical guarantees. On the other hand, pixel-based analysis methods often produce results that are difficult to interpret, with point-wise differential detections that cannot be generalized to the identification of differential regions. The hicream method is entirely data-driven, as it defines candidate regions using clustering. Then, using the tests results provided by diffHiC, it relies on post hoc inference to evaluate each candidate region and provide a minimum proportion of differential pixels within each, thereby identifying the most differential regions. The effectiveness of the method was demonstrated on both semi-simulated data and real data, in relation to external biological signals. Furthermore, by comparing our results to structure-based differential analysis results, we were able to demonstrate that we could identify differential clusters that correspond to differential structures in Hi-C data, such as TADs (Topologically Associating Domains) or compartments. Finally, the interpretation of the detected differential clusters was facilitated by the development of an interactive visualization tool, which also highlighted the shape diversity.